November 2005

PROFIBUS FAM6K

COMPANY CONFIDENTIAL

Table of Contents

1. OVERVIEW 3

1.1 DESCRIPTION 3
1.2 PROFIBUS-DP 4

2. BUS COMMUNICATION 5

2.1 DATA EXCHANGE 5
2.2 GENIUS™ SETTINGS 6
2.2.1 Bus Communication Parameters 6
2.3 DATALOGIC FLOW CONTROL MODE (FCM) 8
2.3.1 Flow Control drivers 10
2.3.2 FCM with DAD Driver 11
2.3.2.1 Control Field 12
2.3.2.2 SAP Field 13
2.3.2.3 Length Field 13
2.3.2.4 Data Transmission from DS6X00 to PLC 14
2.3.2.5 Data Transmission from PLC to DS6X00 17
2.3.2.6 Resynchronisation 18
2.3.2.7 Fragmentation and Reassembling 20
2.3.2.8 SAP Services 22
2.3.2.9 DAD internal queues 22
2.3.3 FCM with DPD Driver 23
2.3.3.1 Control Field 24
2.4 DATA CONSISTENCY 25
2.4.1 Data Consistency with DAD Driver 25
2.4.2 Data Consistency with DPD Driver 27

3. DIGITAL I/O CONDITIONING 28

3.1 DIGITAL INPUT CONDITIONING 29
3.1.1 Phase Echo 30
3.2 DIGITAL OUTPUT CONDITIONING 31
3.2.1 Phase Trigger 31
3.3 READING PHASE VIA PROFIBUS 32

4. NETWORK CONFIGURATION 33

4.1 GSD FILE 33
4.2 GSD INSTALLATION 34
4.3 SCANNER PROGRAMMING VIA GSD FILE 36
4.3.1 Project Modules 36

5. EXAMPLE APPLICATION 41

5.1 INSTALLING THE SCANNER TO THE NETWORK 42
5.2 LOADING THE DEMO APPLICATION PROGRAM 42
5.3 NETWORK CONFIGURATION DESCRIPTION 45
5.3.1 Block Descriptions 47
5.4 EXAMPLE OF DATA EXCHANGE 50

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

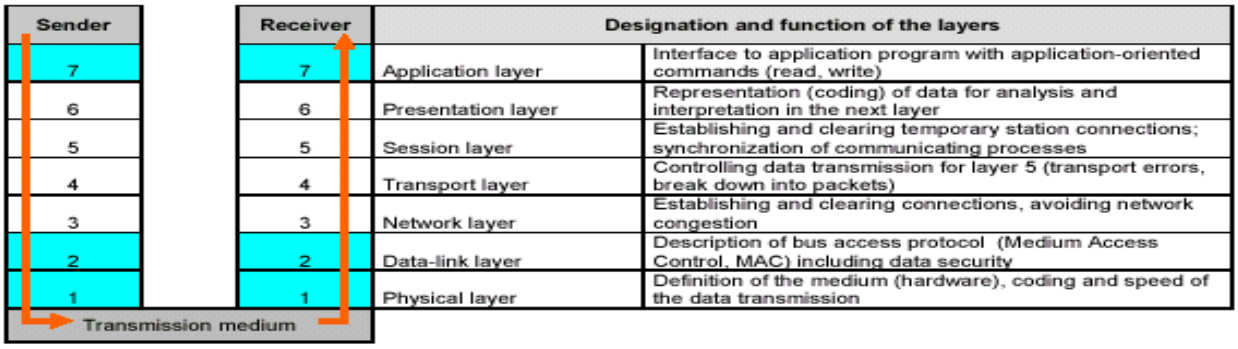
1. Overview

1.1 Description

Profibus is the world's most popular FieldBus.

Profibus is the most widely accepted international networking standard. Nearly universal in Europe and also popular in North America, South America and parts of Africa and Asia. Profibus can handle large amounts of data at high speed and serve the needs of the majority of automation applications.

Profibus was created under German Government leadership in co-operation with automation manufacturers (Siemens) in 1989. Today it is commonly found in Process Control, large assembly and material handling machines. Just a single-cable which is able to wire multi-input sensor blocks, pneumatic valves, complex intelligent devices, smaller sub-networks, operator interfaces and many other devices.



7	7	Application layer	Interface to application program with application-oriented commands (read, write)
6	6	Presentation layer	Representation (coding) of data for analysis and interpretation in the next layer
5	5	Session layer	Establishing and clearing temporary station connections; synchronization of communicating processes
4	4	Transport layer	Controlling data transmission for layer 5 (transport errors, break down into packets)
3	3	Network layer	Establishing and clearing connections, avoiding network congestion
2	2	Data-link layer	Description of bus access protocol (Medium Access Control, MAC) including data security
1	1	Physical layer	Definition of the medium (hardware), coding and speed of the data transmission

The ISO/OSI reference model describes communications between the stations of a communication system: if a communication system does not require some specific functions, the corresponding layers have no purpose and are bypassed. Profibus uses layers 1, 2 and 7.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

1.2 Profibus-DP

Basically Profibus is available in different versions:

- **Profibus-DP** (Decentralized Periphery)
  Multiple masters are possible with Profibus-DP, in which case each slave device is assigned to one master. This means that multiple masters can read inputs from the device but only one master can write outputs to that device.

- Profibus-FMS
  It is a peer to peer messaging format, which allows masters to communicate with one another. Just as in Profibus-DP, up to 126 nodes are available and all can be masters if desired. FMS messages consume more overhead than DP messages.

- Profibus-PA
  PA protocol is the same as the latest Profibus-DP except that voltage and current levels are reduced to meet the requirements of intrinsic safety (Class I div. II) for the process industry.

Datalogic supports **Profibus-DP only**, since this version has been specifically designed for factory automation. System version must be of this type.

Main features:

- Maximum Number of Nodes: 126
- Distance: 100m to 24 Km (with repeaters and fibre optic transmission)
- Baud rate: 9600 to 12M bps
- Messaging formats: Polling, Peer-to-Peer

From here on we will refer to Profibus-DP only.

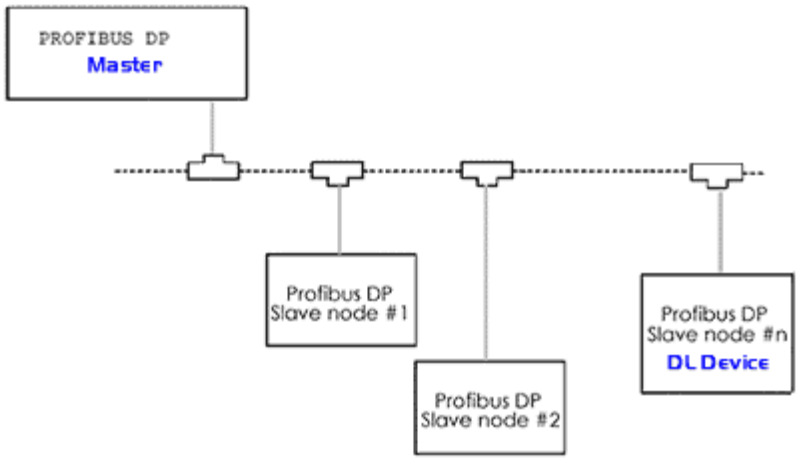
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2. Bus Communication

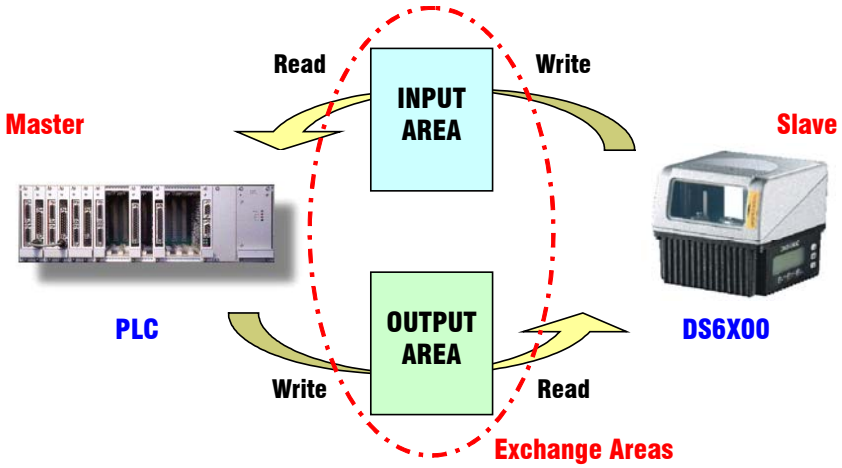
2.1 Data Exchange

Master Profibus is usually a PLC (Siemens S7 or others). Sometimes it could be a PC-based device as well.

DS6X00-10X-012 device is always ***Slave*** in the Profibus network.



Basically two shared memory areas (Exchange Areas) exist between SLAVE and MASTER so both devices provide information to each other. Exchange areas are physically placed on the DS6X00 Profibus card.



Input and output areas always refer to the Master: this means that the scanner writes to the Input buffer and the PLC writes to the Output buffer.

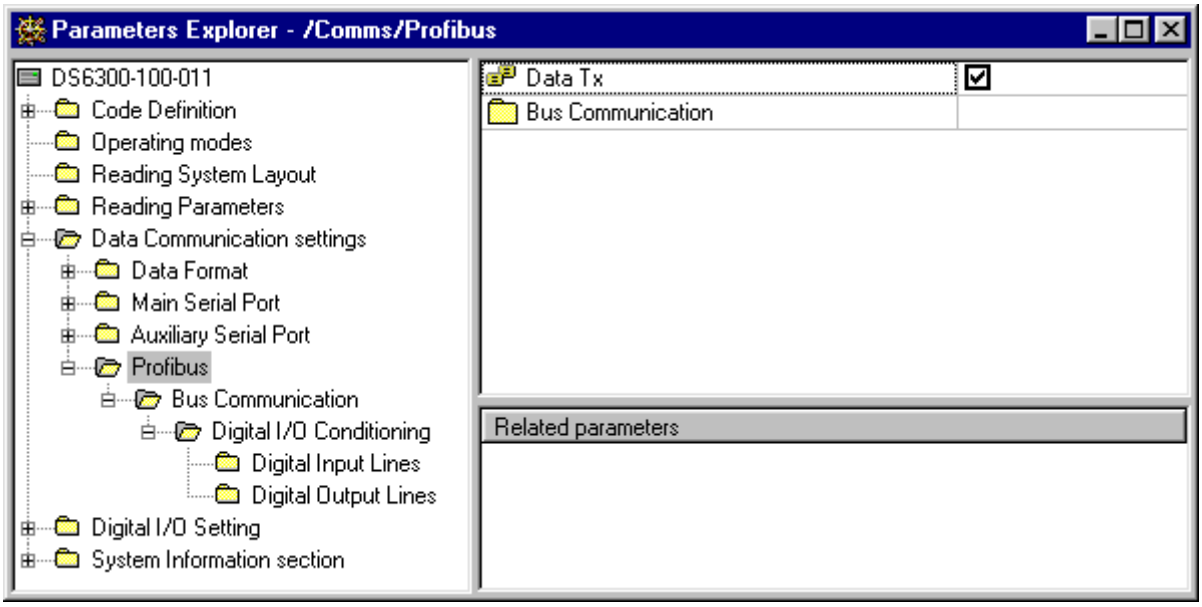
Dimensions of exchange areas can be set to different values by the PLC through the GSD file: the built-in Profibus models DS6300 / DS6400 / DS6500 allow up to **120 bytes for the Input Area** and **32 bytes for the Output Area**.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.2 Genius™ Settings

The Genius™ SW Configuration Tool allows complete parameterisation of Profibus communication.

In order to enable Profibus connectivity, the Data Tx parameter must be selected first (as default):



2.2.1 Bus Communication Parameters

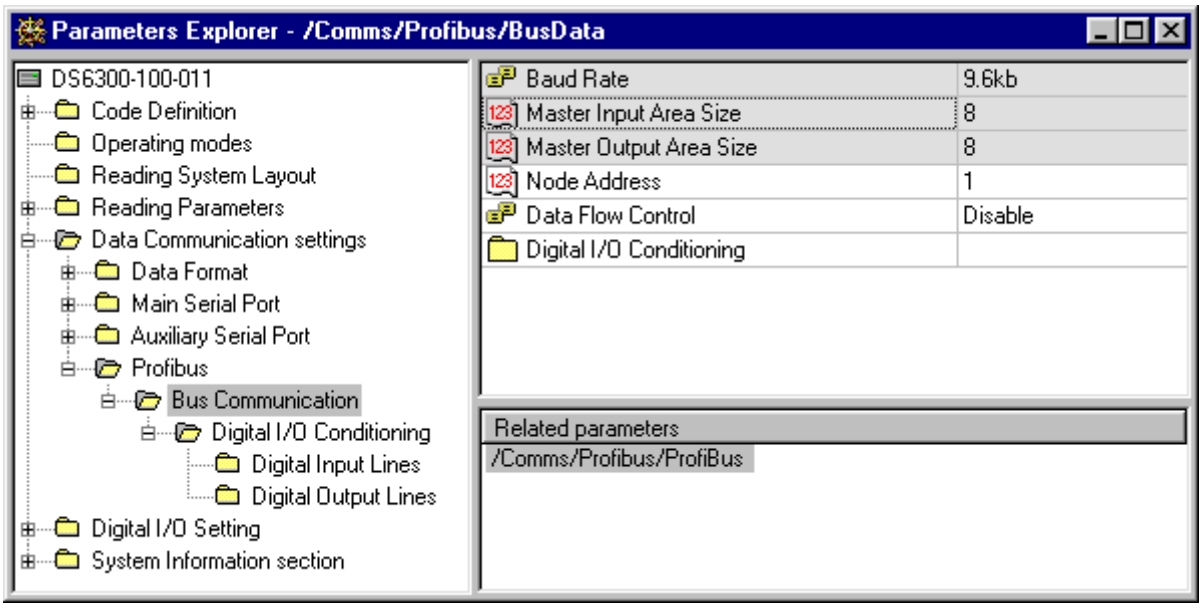
The Slave is identified over the Profibus network by setting its own address (Node Address parameter). This value, which is unique for each slave, should match the address specified while configuring your network, so that the PLC is able to detect the Slave properly.

Data Flow Control enables a powerful way to manage and optimise communication with the Profibus Master. A dedicated program running on the PLC is required to take advantage of this feature (see details in the "Datalogic Flow Control Mode (FCM)" paragraph).

Final implementation should make use of Datalogic Flow Control to obtain maximum reliability and optimal synchronisation between Master and Slave. As a first approach, Data Flow Control = Disable is the suggested way to check the HW/SW configuration of the Profibus network.

Within the Bus Communication section Baud Rate, Master Input Area Size and Master Output Data Size are read only, as these parameters are decided by the Profibus Master.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.



This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3 Datalogic Flow Control Mode (FCM)

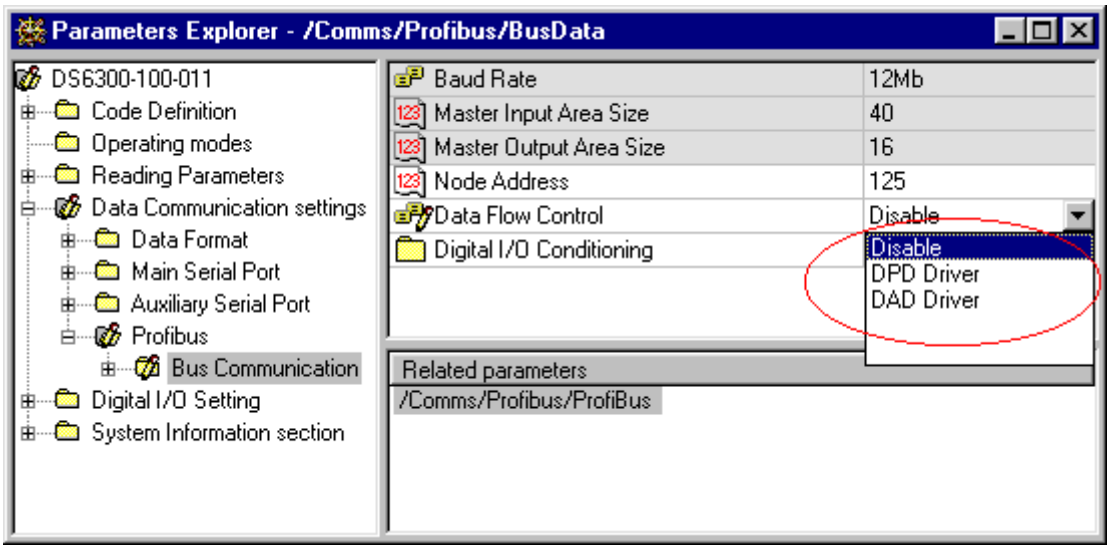
The Datalogic Flow Control Mode is a powerful way to manage and optimise the communication with the Profibus Master. By enabling the FCM a few bytes of the exchange areas are reserved for driver operations and the rest are used by the application layer.

The reserved bytes are used to implement many different features such as:

- Flow-control and corresponding buffering in both directions
- Fragmentation and reassembling of data longer than the exchange area sizes
- Synchronisation of flow control numbers
- Service Access Point oriented communication
- Length information

Note: If the Datalogic Flow Control is disabled, all the bytes of the exchange areas are used by the application layer. The input area is updated whenever a new reading event has to be transferred to the Master station.
→ In this situation, the Master must read the input area before it changes due to a new message, typically new barcode occurrence.
→ Moreover, two occurrences of the same barcode cannot be understood, since the input area does not change.
→ In addition, if application data is longer than input area sizes, data is automatically truncated.

FCM can be selected by means of the Data Flow Control parameter.



Basically three options are available:

- DAD Driver → *FCM compatible with "Flow Control = Anybus" used in new Datalogic devices*
- DPD Driver → *FCM compatible with "Flow Control = Profibus" used in MX4000-1100*
- Disable → *No Datalogic Flow Control*

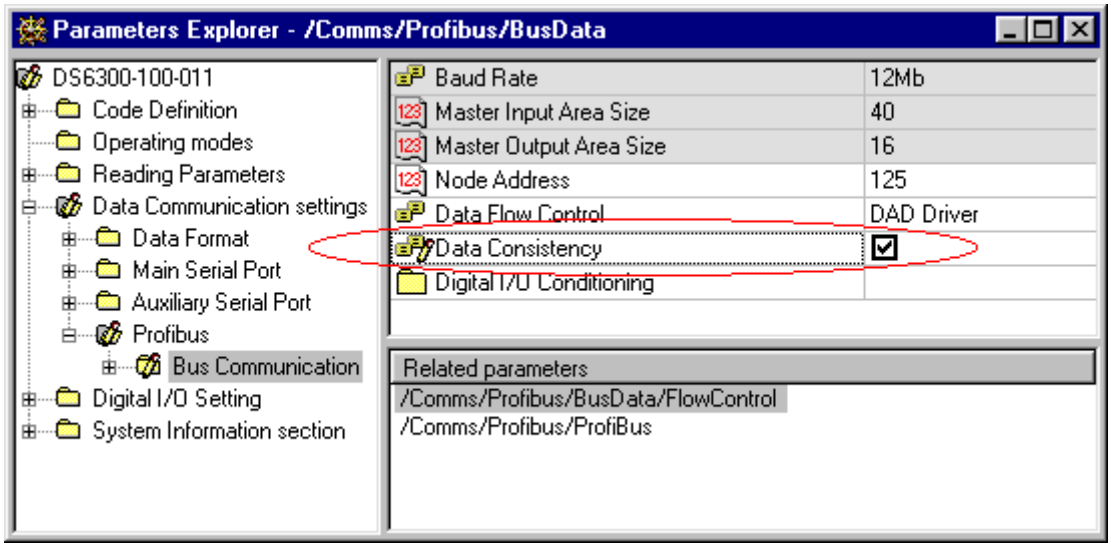
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Although both drivers implement the same above features, two options are currently available to obtain the maximum compatibility towards different Datalogic family devices.

See the cross reference table below about supported FCM:

6K Family	Article Number	**DAD Driver**	**DPD Driver**
DS6300-100-011	931351020	X	X
DS6300-105-011	931351070	X	X
DS6400-100-011	931351095	X	X
DS6400-105-011	931351105	X	X
DS6500-100-011	931401004	X	X
DS6500-200-011	931401005	X	X
DS6500-105-011	931401014	X	X
DS6500-205-011	931401015	X	X
CBOX Family	Article Number	**DAD Driver**	**DPD Driver**
CBOX-300 Profibus-DP	93A301000	X	X
CBOX-310 Profibus-DP with Display	93A301030	X	X
MX Family			
MX4000-1100	935151010		X
MX4000-1100 SB2498 12MB/s	935151040	X	

When FCM is utilised, the Data Consistency option can be enabled to improve the overall safety in data communication. This option is mandatory if Master Input Area size exceeds 32 bytes.
See details in the next paragraphs.

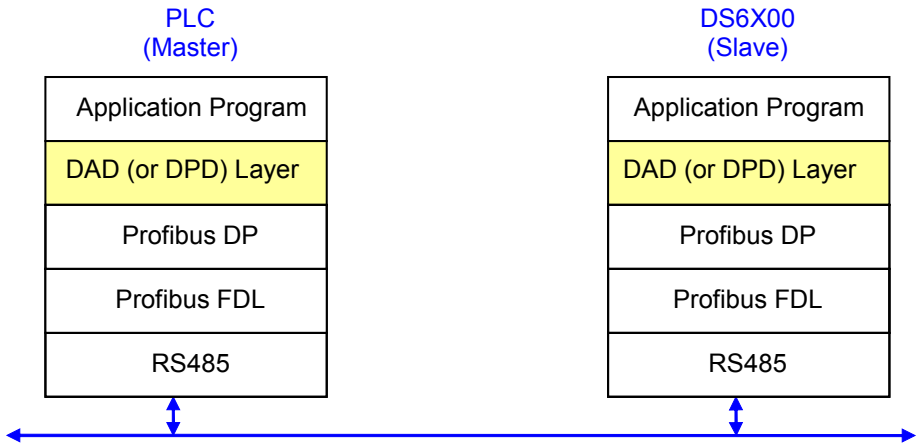


This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.1 Flow Control drivers

The Flow Control driver is a layer that is built upon the intrinsic DP data exchange mechanism. Basically such a layer is required because the intrinsic DP Profibus mechanism is not message oriented.

In the following figure the complete Stack is represented:



Two Flow Control drivers are available:

- DAD Driver → Compatible with *Flow Control = Anybus*
- DPD Driver → Compatible with *Flow Control = Profibus*

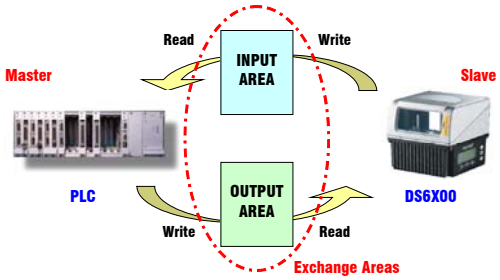
As described previously, they can be selected separately on DS6X00 by means of the Genius™ configuration tool. Obviously, the corresponding driver must be implemented on the PLC side.

The DAD driver should be the preferred solution for brand new installations.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2 FCM with DAD Driver

In order to implement the flow controlled version of the driver, exchange areas must be congruently compiled in both directions.



From now on we refer to the Input Area as a buffer made up of InputAreaSize bytes:

IN[0]	IN[1]	IN[2]	…	IN[InputAreaSize - 1]

and to the Output Area as a buffer made up of OutputAreaSize bytes:

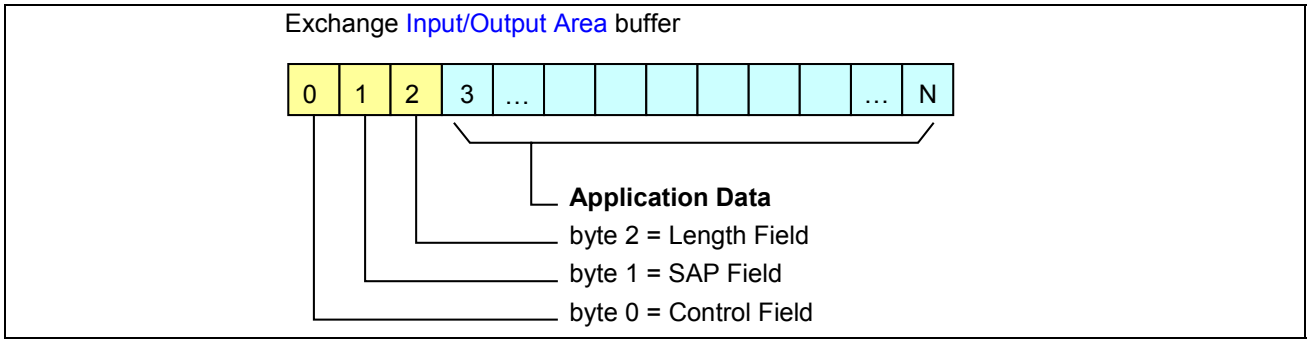
OUT[0]	OUT[1]	OUT[2]	…	OUT[OutputAreaSize - 1]

Only the first three bytes are used by the DAD Driver layer in both buffers:

Control Field (byte 0) — used to issue and control the driver primitives such as flow-control, fragmentation and resynchronisation.

Service Access Point Field (byte 1) — used to distinguish among different services and to provide future expandability. (Since this SAP definition is introduced by the DAD Driver, it must not be confused with the SAP that is defined by the international standard).

Length Field (byte 2) — contains the number of bytes used by the application layer:
Length Field ≤ (InputAreaSize – 3) for the Input Area
Length Field ≤ (OutputAreaSize – 3) for the Output Area.



The Application Data buffer holds useful information, typically the barcode messages, processed by the application program. IN[3] contains the first significant byte of the Application Data buffer (the same first byte you would see if DS6X00 transmitted the barcode buffer onto the Auxiliary port instead of the Profibus interface).

The structure of the application buffer and its length strictly depend on the selected data format on the DS6X00. Barcode messages longer than (InputAreaSize – 3) will be split in pieces through an automatic fragmentation process (see details in the "Fragmentation and Reassembling" paragraph).

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2.1 Control Field

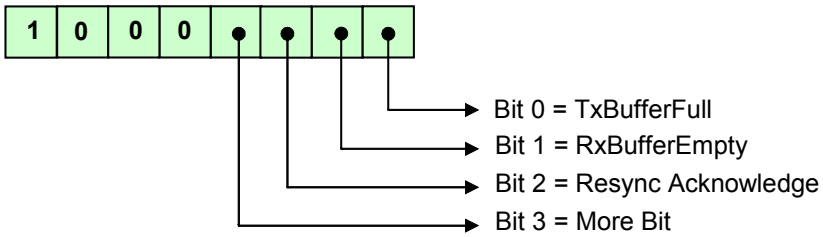
This is the core of the flow controlled communication.

The Input Area structure reserves bit 0 and bit 1 of IN[0] for handshake purposes while the Output Area structure, which is symmetrical, reserves bit 0 and 1 of OUT[0].

At any time the Master station can make a resynchronization request by means of bit 2 of the Output Area. This process, which resets the synchronization numbers (bit 0 and bit 1 of both Input and Output areas), has to be acknowledged by the Slave on bit 2 of the Input Area.

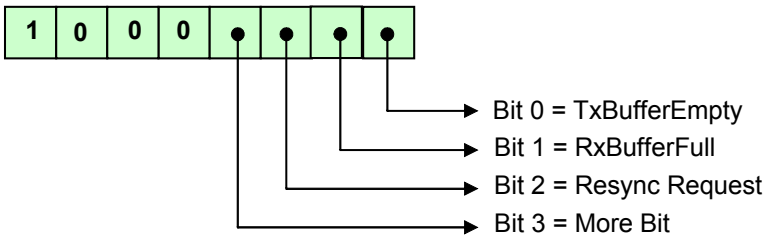
Bit 3 is used to control a fragmentation sequence in both directions.

Control Field byte of Input Area → IN[0]



IN[0].bit0	**TxBufferFull** toggles when Slave has made available new Input Area data
IN[0].bit1	**RxBufferEmpty** toggles when Output Area data has been read by Slave
IN[0].bit2	**Resync Acknowledge** set to 1 as an acknowledge to a resync request. With this bit, the master can detect a slave is on line.
IN[0].bit3	**More Bit** is 1 when this is not the last piece of a fragmentation sequence while it is 0 when this is the last piece
IN[0].bit4,5,6,7	Set to **0**, **0**, **0**, **1** when DAD messaging protocol is used

Control Field byte of Output Area → OUT[0]



OUT[0].bit0	**TxBufferEmpty** must toggle when Input Area data has been read by Master
OUT[0].bit1	**RxBufferFull** must toggle when Master makes available new Output Area data
OUT[0].bit2	**Resync Request** set to 1 for one second to resynchronize the slave. After resynchronization, all 4 handshake bits are set to 0
OUT[0].bit3	**More Bit** must be 1 when this is not the last piece of a fragmentation sequence and it must be 0 when this is the last
OUT[0].bit4,5,6,7	Set to **0**, **0**, **0**, **1** when DAD messaging protocol is used

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2.2 SAP Field

SAP (Service Access Point) is an identifier that is used to implement multiple services sharing the same communication channel between two remote stations.

The following values have been defined:

- SAP = 0 Used to transfer information messages between DS6X00 and PLC
- SAP = 255 Reserved for driver services (see details in the "SAP Services" paragraph)

All other SAP values are free and they could be used by dedicated application programs after agreement between the application programs themselves.

2.3.2.3 Length Field

The Application layer uses all or a part of the remaining bytes of the Exchange Area buffers that are not used by the DAD Driver. The Length Field is introduced to keep the information of how many bytes are really used by the Application Layer.

A fragment that is not the last one of a fragmentation sequence must fill this field with [InputAreaSize – 3] (or [OutputAreaSize - 3]), depending on whether it is an Input/Output fragment. Otherwise this field gets a value that is less than or equal to [InputAreaSize – 3].

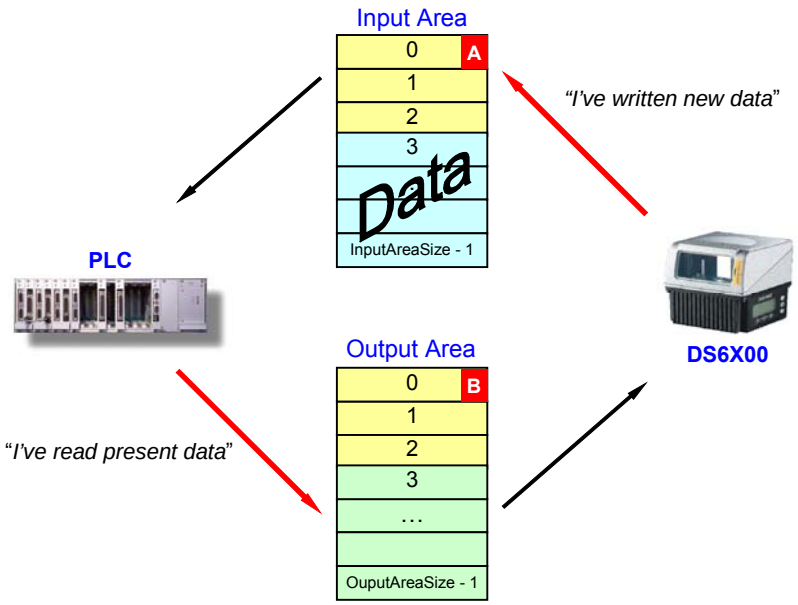
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2.4 Data Transmission from DS6X00 to PLC

This paragraph describes how it is possible to exchange messages with flow control. The communication mechanism is simple:

- **IN[0].bit0** [A] is used by DS6X00 to notify that "*Slave has written a new data so Master can read it*"
- **OUT[0].bit0** [B] must be used by PLC to notify that "*Master has read last data so Slave can send next message*"

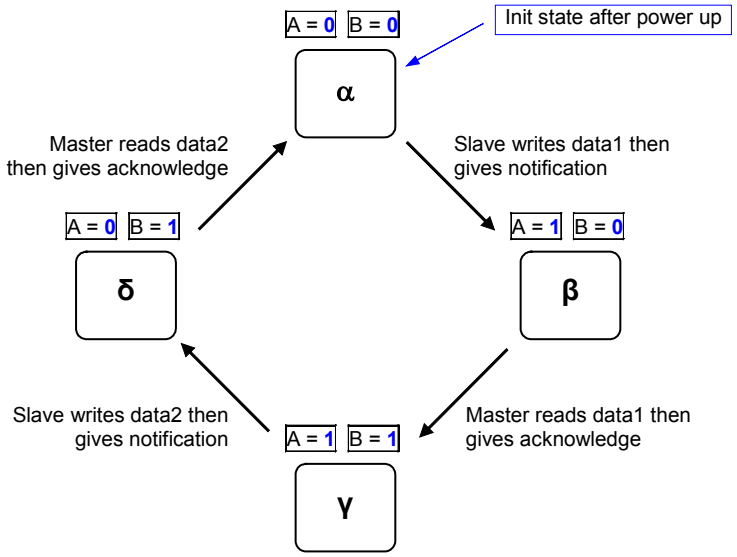
This happens each time bit A (or B) changes its state (toggles). Bit level doesn't matter, only the transition has to be considered.



The following state machine shows data transmission from Slave to Master. Please note that each cycle transfers two data messages.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

A indicates IN[0].bit0 [TxBufferFull]
B indicates OUT[0].bit0 [TxBufferEmpty]



Let's analyse a typical data exchange based on the following settings:

- Flow Control = **DAD Driver**
- Input Area Size = **16**
- Output Area Size = **8**

- After power up Input and Output areas are generally filled by zero. According to DAD driver implementation, Input area has Control Field = 80_{Hex} and SAP = 00_{Hex}.

Input Area	**80**$_{Hex}$ **00**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- Also PLC must set the Control Field of Output area properly, as long as DAD messaging protocol is utilised.

Input Area	80$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	**80**$_{Hex}$ **00**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- DS6X00 reads a barcode "123456". Let's assume standard data formatting with <STX> as header and <CR><LF> as terminators. DS6X00 toggles bit **A**.

Input Area	**81**$_{Hex}$ 00$_{Hex}$ **09**$_{Hex}$ **02**$_{Hex}$ **31**$_{Hex}$ **32**$_{Hex}$ **33**$_{Hex}$ **34**$_{Hex}$ **35**$_{Hex}$ **36**$_{Hex}$ **0D**$_{Hex}$ **0A**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	80$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- PLC detects transition of bit **A** so now it can read incoming data (it copies 9 bytes in its memory from IN[3] on) then toggles bit **B** as acknowledge.
  ***Note**: before the acknowledge, all further barcodes read by DS6X00 are buffered.*

Input Area	81$_{Hex}$ 00$_{Hex}$ 09$_{Hex}$ 02$_{Hex}$ 31$_{Hex}$ 32$_{Hex}$ 33$_{Hex}$ 34$_{Hex}$ 35$_{Hex}$ 36$_{Hex}$ 0D$_{Hex}$ 0A$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	**81**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- DS6X00 reads a barcode "10DL" and toggles bit **A**.

Input Area	**80**$_{Hex}$ 00$_{Hex}$ **07**$_{Hex}$ **02**$_{Hex}$ **31**$_{Hex}$ **30**$_{Hex}$ **44**$_{Hex}$ **4C**$_{Hex}$ **0D**$_{Hex}$ **0A**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	81$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

- PLC reads new data message (it copies 7 bytes in its memory from IN[3] on) then toggles bit **B** as acknowledge.

Input Area	80_{Hex} 00_{Hex} 07_{Hex} 02_{Hex} 31_{Hex} 30_{Hex} 44_{Hex} $4C_{Hex}$ $0D_{Hex}$ $0A_{Hex}$ 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex}
Output Area	**80_{Hex}** 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex}

- DS6X00 performs *No Read* and toggles bit **A**. Let's assume <CAN> as *Global No Read* character.

Input Area	**81_{Hex}** 00_{Hex} **04_{Hex}** **02_{Hex}** **18_{Hex}** **$0D_{Hex}$** **$0A_{Hex}$** 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex}
Output Area	80_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex}

- PLC reads new data message (it copies 4 bytes in its memory from IN[3] on) then toggles bit **B** as acknowledge.

Input Area	81_{Hex} 00_{Hex} 04_{Hex} 02_{Hex} 18_{Hex} $0D_{Hex}$ $0A_{Hex}$ 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex}
Output Area	**81_{Hex}** 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex} 00_{Hex}

- Data exchange continues…

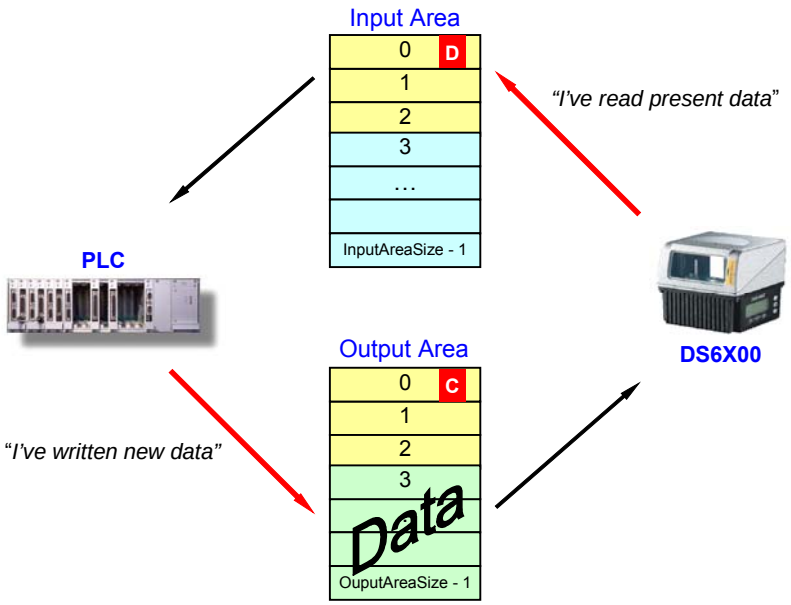
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2.5 Data Transmission from PLC to DS6X00

Analogous to the previous paragraph, flow control works even when data are coming from the Master towards the Slave. The communication mechanism is based on the same concepts:

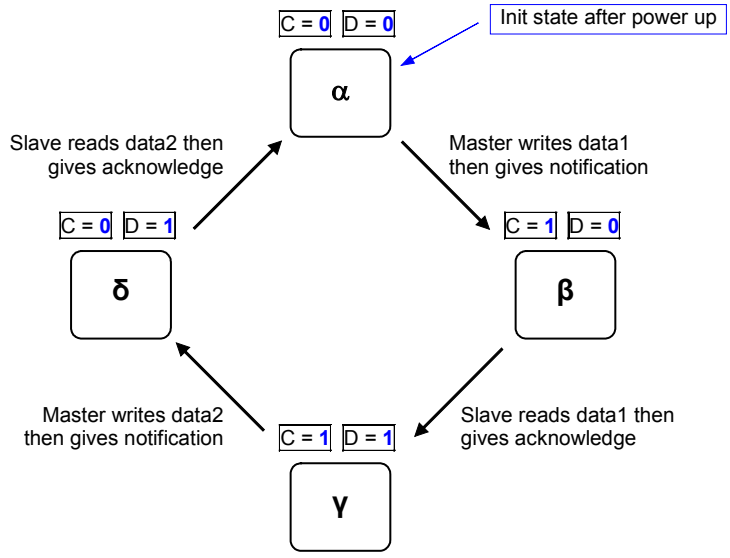
- **OUT[0].bit1** [**C**] must be used by PLC to notify that "*Master has written new data so Slave can read it*"
- **IN[0].bit1** [**D**] is used by DS6X00 to notify that "*Slave has read data so Master can send next message*"

This happens each time bit C (or D) changes its state (toggles). Bit level doesn't matter, only the transition has to be considered.



The following state machine shows data transmission from Master to Slave. Please note that each cycle transfers two data messages.

C indicates OUT[0].bit1 [RxBufferFull]
D indicates IN[0].bit1 [RxBufferEmpty]


C = 0 D = 0
Init state after power up
α
Slave reads data2 then gives acknowledge
Master writes data1 then gives notification
C = 0 D = 1
C = 1 D = 0
δ
β
Master writes data2 then gives notification
C = 1 D = 1
Slave reads data1 then gives acknowledge
γ

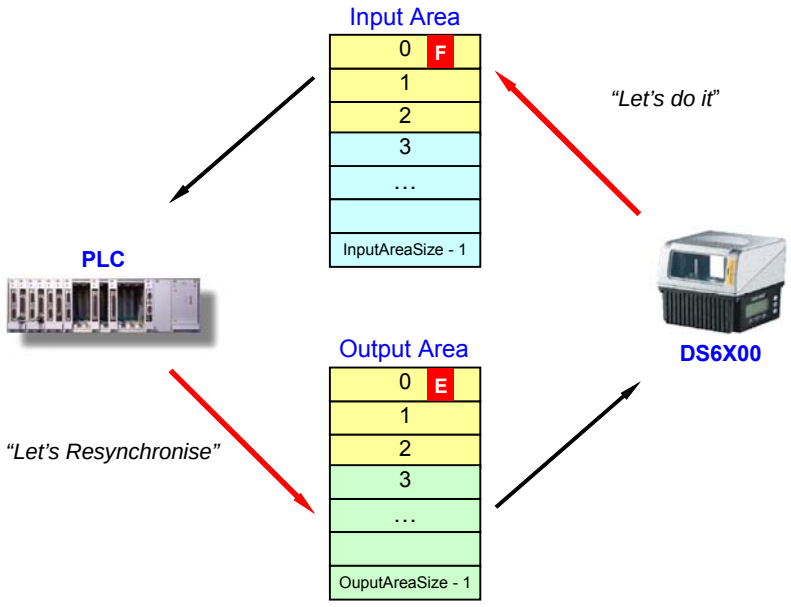
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2.6 Resynchronisation

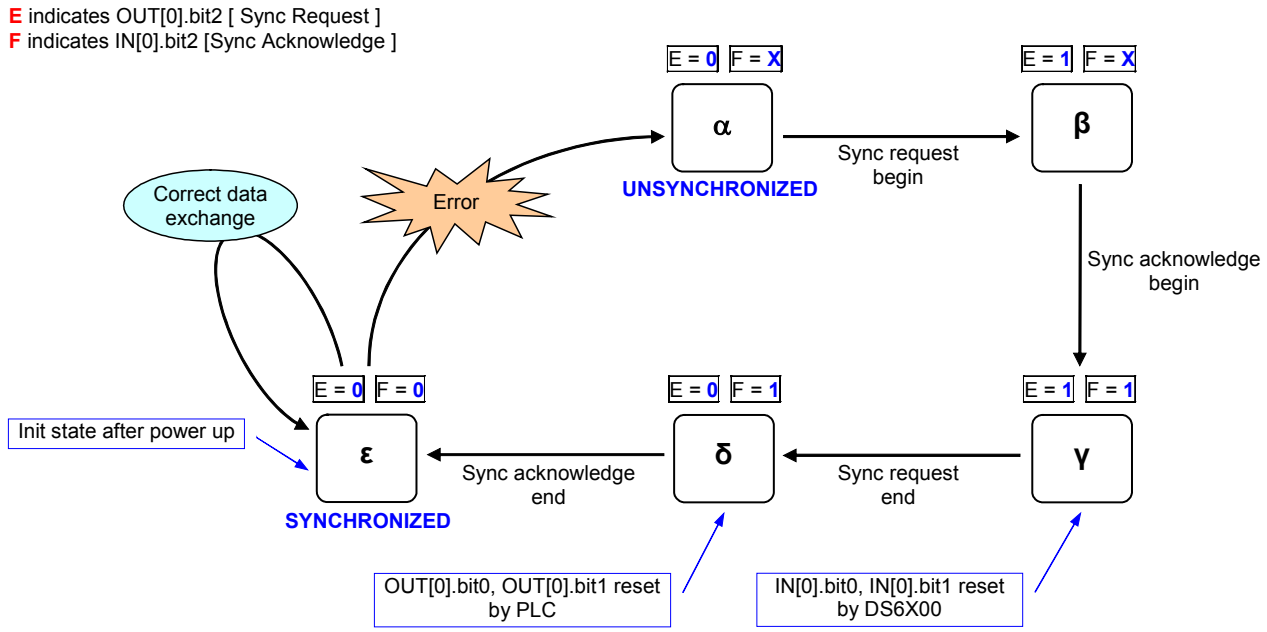
The resynchronisation process restarts the messaging protocol from a predefined state.
It may be used either at the Master startup to detect if the Slave is on line or during normal operations in case of errors requiring a protocol reset procedure.

The process is based on bit two:

- **OUT[0].bit2** [E] must be used by PLC to request the Resynchronisation
- **IN[0].bit2** [F] is used by DS6X00 to acknowledge the request



The following state machine shows the resynchronisation cycle, requested by the PLC and performed together with DS6X00:



This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Let's analyse the resynchronisation process, starting from the previous data exchange discussed in "Data Transmission from DS6X00 to PLC"…

Input Area	81_{Hex}	00_{Hex}	04_{Hex}	02_{Hex}	18_{Hex}	$0D_{Hex}$	$0A_{Hex}$	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}
Output Area	81_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}							

- PLC requests resynchonisation by setting bit **E** = 1.

Input Area	81_{Hex}	00_{Hex}	04_{Hex}	02_{Hex}	18_{Hex}	$0D_{Hex}$	$0A_{Hex}$	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}
Output Area	**85_{Hex}**	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}							

- DS6X00 detects the request and so it resets **IN[0].bit0** and **IN[0].bit1**. Then it gives an acknowledge back to the PLC by means of bit **F**.

Input Area	**84_{Hex}**	00_{Hex}	04_{Hex}	02_{Hex}	18_{Hex}	$0D_{Hex}$	$0A_{Hex}$	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}
Output Area	85_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}							

- PLC has to reset **OUT[0].bit0** and **OUT[0].bit1** before completing its request with bit **E** = 0.

Input Area	84_{Hex}	00_{Hex}	04_{Hex}	02_{Hex}	18_{Hex}	$0D_{Hex}$	$0A_{Hex}$	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}
Output Area	**80_{Hex}**	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}							

- DS6X00 completes the acknowledge process by setting bit **F** = 0.

Input Area	**80_{Hex}**	00_{Hex}	04_{Hex}	02_{Hex}	18_{Hex}	$0D_{Hex}$	$0A_{Hex}$	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}
Output Area	80_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}	00_{Hex}							

- Now Flow Control has been returned to a predefined state. All data exchange bits in the Control Field are surely zero and data transmission can proceed safely.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2.7 Fragmentation and Reassembling

The fragmentation process is activated whenever Application Data cannot be contained in the related exchange area. Basically long messages are split into pieces which are transmitted separately. Reassembling allows the reconstruction of the whole messages.

DS6X00 already implements these functions in the DAD layer, while the PLC needs a congruent management.

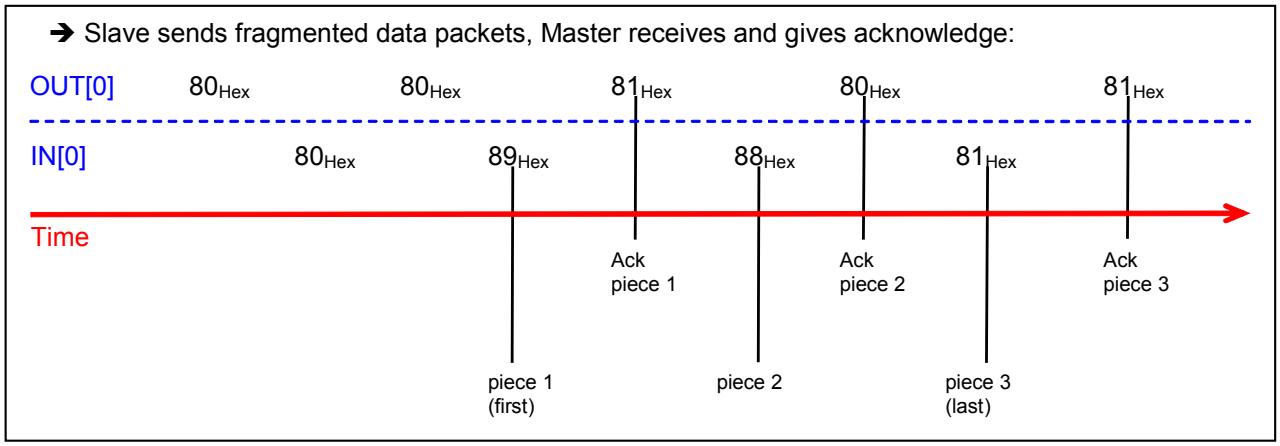
The fragmentation is based on the **More Bit** (bit 3) in the Control Field byte.

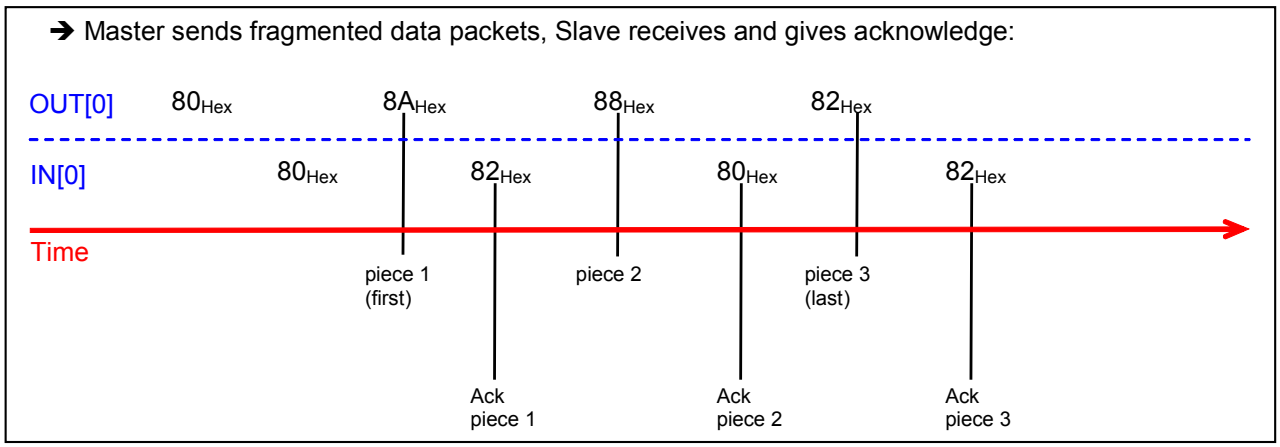
More Bit = 0 indicates that all the information is included within the current message. When Application Data is longer than (exchange area size – 3), the first partial message is transmitted having More Bit = 1. Following fragments keep More Bit = 1 and only the last piece will have More Bit = 0 again. Thanks to this mechanism, the receiver station may detect the last piece and so reassemble the entire information.

Some notes:

- DS6X00 can manage application messages up to 256 bytes
- Intermediate fragments have Length Field = (exchange area size – 3)
- Last fragment has Length Field ≤ (exchange area size – 3)
- Bit0 and bit1 of both Input and Output areas are independently managed for any fragment

The following figures show how the Control Byte changes according to the fragmentation process. Both data flow directions are considered.





This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Let's analyse a fragmented data exchange based on the following settings:

- Flow Control = **DAD Driver**
- Input Area Size = **16**
- Output Area Size = **8**

- After power up Input and Output areas are generally filled by zero. According to DAD driver implementation, Input area has Control Field = 80_{Hex} and SAP = 00_{Hex}.

Input Area	**80**$_{Hex}$ **00**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- Also PLC must set the Control Field of Output area properly, as long as DAD messaging protocol is utilised.

Input Area	80$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	**80**$_{Hex}$ **00**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- DS6X00 reads a barcode with content "*1234567890abcde1234567890abcde*". Let's assume standard data formatting with <STX> as header and <CR><LF> as terminators. In this condition since the whole message cannot be included in Input Area, DS6X00 transmits first fragment "<STX>*1234567890ab*" only (setting More Bit = 1) then it toggles bit **A**.

Input Area	**89**$_{Hex}$ 00$_{Hex}$ **0D**$_{Hex}$ **02**$_{Hex}$ **31**$_{Hex}$ **32**$_{Hex}$ **33**$_{Hex}$ **34**$_{Hex}$ **35**$_{Hex}$ **36**$_{Hex}$ **37**$_{Hex}$ **38**$_{Hex}$ **39**$_{Hex}$ **30**$_{Hex}$ **61**$_{Hex}$ **62**$_{Hex}$
Output Area	80$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- PLC detects transition of bit **A** so now it can read first incoming fragment (it copies 13 bytes in its memory from IN[3] on) then toggles bit **B** as acknowledge.

Input Area	89$_{Hex}$ 00$_{Hex}$ 0D$_{Hex}$ 02$_{Hex}$ 31$_{Hex}$ 32$_{Hex}$ 33$_{Hex}$ 34$_{Hex}$ 35$_{Hex}$ 36$_{Hex}$ 37$_{Hex}$ 38$_{Hex}$ 39$_{Hex}$ 30$_{Hex}$ 61$_{Hex}$ 62$_{Hex}$
Output Area	**81**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- DS6X00 detects transition of bit **B** so it sends second fragment "*cde1234567890*" (still More Bit = 1) and toggles bit **A**.

Input Area	**88**$_{Hex}$ 00$_{Hex}$ **0D**$_{Hex}$ **63**$_{Hex}$ **64**$_{Hex}$ **65**$_{Hex}$ **31**$_{Hex}$ **32**$_{Hex}$ **33**$_{Hex}$ **34**$_{Hex}$ **35**$_{Hex}$ **36**$_{Hex}$ **37**$_{Hex}$ **38**$_{Hex}$ **39**$_{Hex}$ **30**$_{Hex}$
Output Area	81$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- PLC reads second fragment (it copies 13 bytes in its memory from IN[3] on) then toggles bit **B** as acknowledge.

Input Area	88$_{Hex}$ 00$_{Hex}$ 0D$_{Hex}$ 63$_{Hex}$ 64$_{Hex}$ 65$_{Hex}$ 31$_{Hex}$ 32$_{Hex}$ 33$_{Hex}$ 34$_{Hex}$ 35$_{Hex}$ 36$_{Hex}$ 37$_{Hex}$ 38$_{Hex}$ 39$_{Hex}$ 30$_{Hex}$
Output Area	**80**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- DS6X00 sends third (last) fragment "*abcde<CR><LF>*" (finally More Bit = 0) and toggles bit **A**.

Input Area	**81**$_{Hex}$ 00$_{Hex}$ **07**$_{Hex}$ **61**$_{Hex}$ **62**$_{Hex}$ **63**$_{Hex}$ **64**$_{Hex}$ **65**$_{Hex}$ **0D**$_{Hex}$ **0A**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	80$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- PLC reads last fragment (it copies 7 bytes in its memory from IN[3] on) and now the reassembling can be completed. Then it toggles bit **B** as acknowledge.

Input Area	81$_{Hex}$ 00$_{Hex}$ 07$_{Hex}$ 61$_{Hex}$ 62$_{Hex}$ 63$_{Hex}$ 64$_{Hex}$ 65$_{Hex}$ 0D$_{Hex}$ 0A$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$
Output Area	**81**$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$ 00$_{Hex}$

- Whole message has been completely transmitted.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.2.8 SAP Services

FLUSH QUEUE is the unique driver service currently available. It performs flushing of the internal queues and may be issued at any time.

FLUSH QUEUE Service

Request: Flush data buffers (issued by the Master station to DS6X00)
Action: Flush all information from previous decoding phases
Response: Command accepted / Command rejected (generated by DS6X00 toward Master)

Application data areas must be formatted as follows:

Request Command	**byte 3**	**byte 4**
Flush data buffer	**'['** (5B Hex)	**'F'**(46 Hex)

Response Command	**byte 3**	**byte 4**
Command accepted	**'A'** (41 Hex)	**' '**(20 Hex)
Command rejected	**'C'** (43 Hex)	**' '**(20 Hex)

2.3.2.9 DAD internal queues

DS6X00-10X-012 has two internal queues (one for each direction) to keep the application events: input queue and output queue.

The input queue is used when a new message (generally a barcode) has to be transmitted by D6X000 before the Master station has generated all the acknowledge handshakes for each previous transmission.
The output queue is rarely used at the moment.

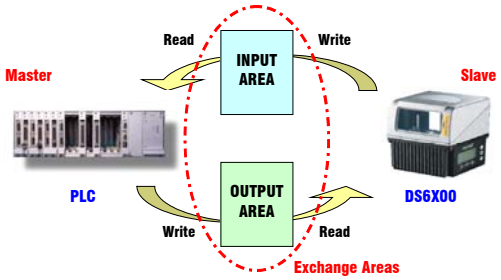
The queues are sized in the following way:
- 50 elements of 120 bytes each for input queue (data flow from Slave to Master)
- 26 elements of 32 bytes each for output queue (data flow from the Master to Slave)

The queues may be flushed by the Master station through the SAP=255 primitive. This is generally done at the Master startup if the Master station wants to cancel all the previous buffers that were generated before its startup. However, the Master station is free to decide not to cancel them.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.3 FCM with DPD Driver

In order to implement the flow controlled version of the driver, exchange areas must be congruently compiled in both directions.

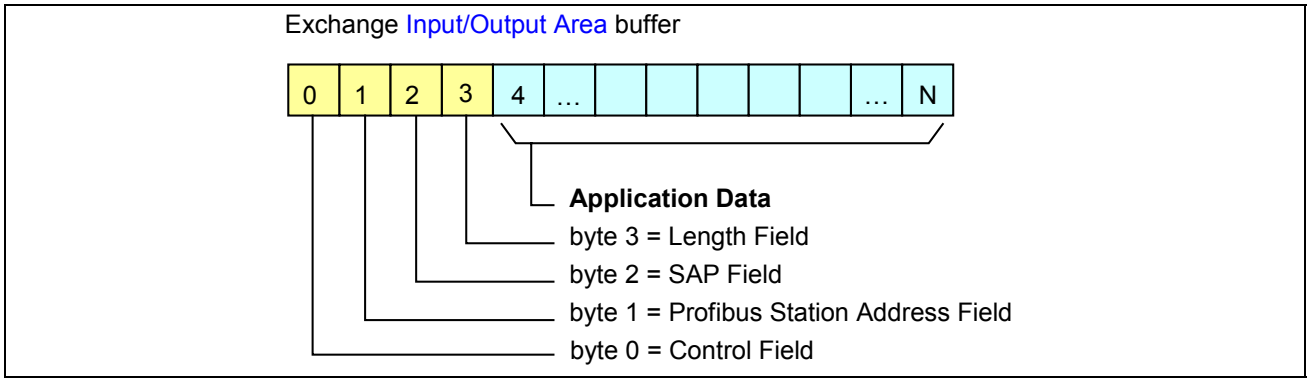


This driver supports all the features listed in the “FCM with DAD Driver” paragraph and implements all the handshake mechanisms previously discussed.

Differently from DAD Driver, DPD implement the further Profibus Station Address Field and Control Field must have bit7 set to zero (see par. 2.3.3.1).

The first **four** bytes are used by the DPD Driver layer in both buffers as follows:

Control Field (byte 0)	used to issue and control the DPD Driver primitives such as flow-control, fragmentation and resynchronisation.
Profibus Station Address Field (byte 1)	contains the information of the selected DS6X00 address of the Profibus network.
Service Access Point Field (byte 2)	used to distinguish among different services and to provide future expandability. (Since this SAP definition is introduced by the DPD Driver, it must not be confused with the SAP that is defined by the international standard).
Length Field (byte 3)	contains the number of bytes used by the application layer: Length Field ≤ (InputAreaSize – 4) for the Input Area Length Field ≤ (OutputAreaSize – 4) for the Output Area.



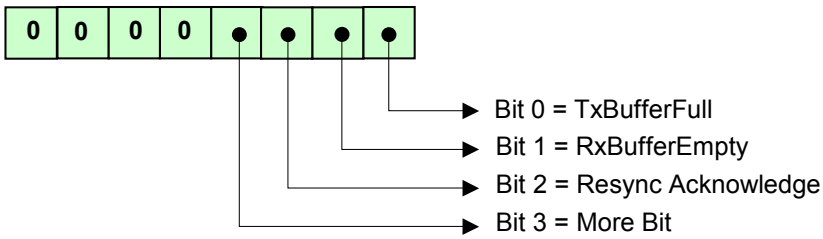
The Application Data buffer holds the information starting from byte IN[4] and the fragmentation process is managed by DS6X00 when barcode messages are longer than (InputAreaSize – 4).

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.3.3.1 Control Field

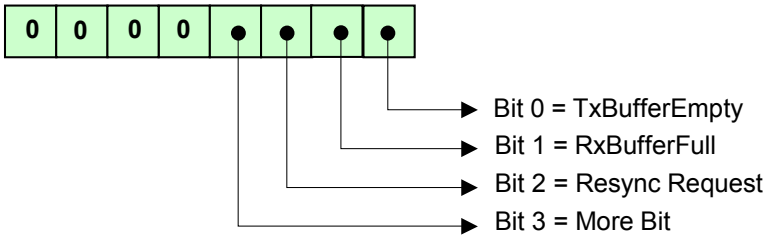
Differently from DAD Driver, just note that Control Field must have bit7 set to zero.

Control Field byte of Input Area → IN[0]



IN[0].bit0	**TxBufferFull** toggles when Slave has made available new Input Area data
IN[0].bit1	**RxBufferEmpty** toggles when Output Area data has been read by Slave
IN[0].bit2	**Resync Acknowledge** set to 1 as an acknowledge to a resync request. With this bit, the master can detect a slave is on line.
IN[0].bit3	**More Bit** is 1 when this is not the last piece of a fragmentation sequence while it is 0 when this is the last piece
IN[0].bit4,5,6,7	Set to **0**, **0**, **0**, **0** when DPD messaging protocol is used

Control Field byte of Output Area → OUT[0]



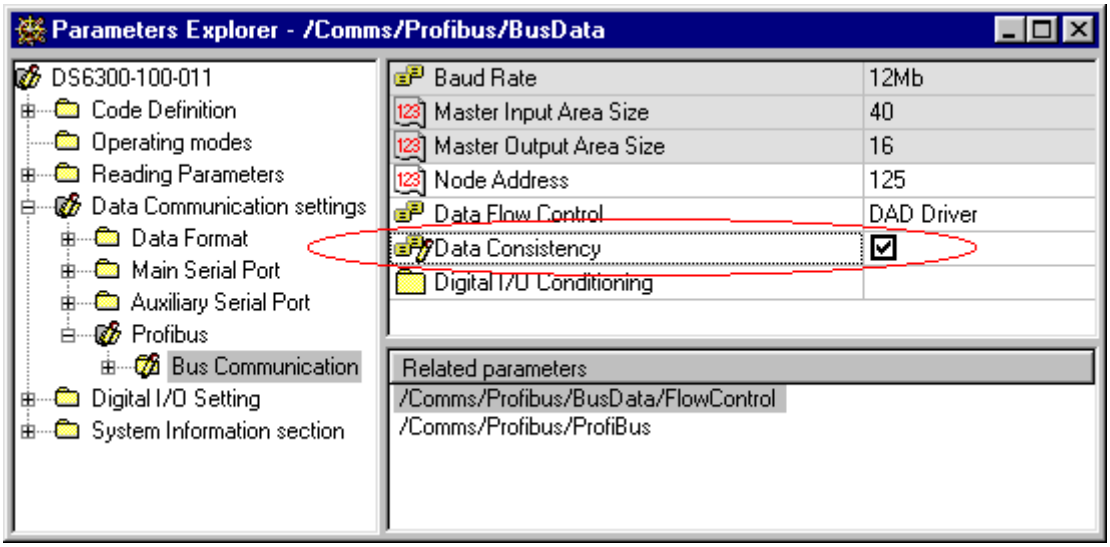
OUT[0].bit0	**TxBufferEmpty** must toggle when Input Area data has been read by Master
OUT[0].bit1	**RxBufferFull** must toggle when Master makes available new Output Area data
OUT[0].bit2	**Resync Request** set to 1 for one second to resynchronize the slave. After resynchronization, all 4 handshake bits are set to 0
OUT[0].bit3	**More Bit** must be 1 when this is not the last piece of a fragmentation sequence and it must be 0 when this is the last
OUT[0].bit4,5,6,7	Set to **0**, **0**, **0**, **0** when DPD messaging protocol is used

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.4 Data Consistency

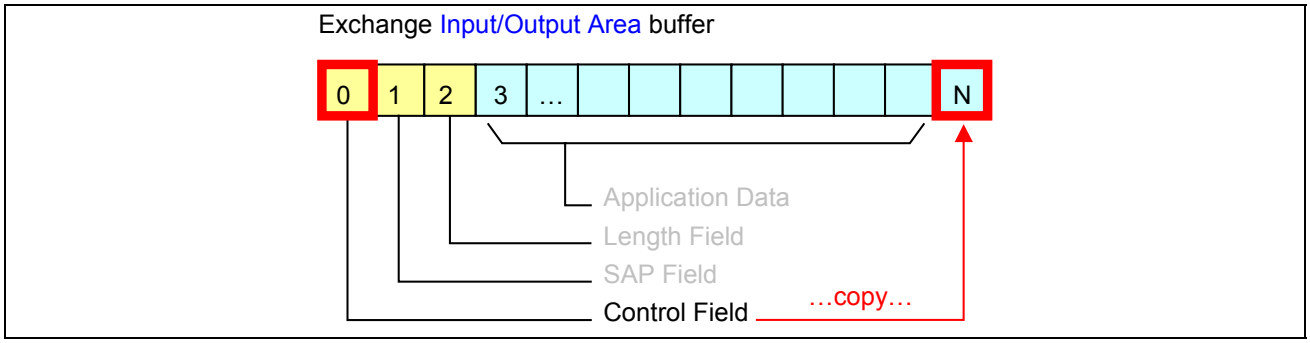
When Flow Control (either DAD or DPD Driver) is used, the Data Consistency option is available.

If enabled, the driver implements a specific mechanism to guarantee consistency of both transmitted and received data over the entire size of the exchange area.



2.4.1 Data Consistency with DAD Driver

As shown in the following figure, the application layer copies the Control Field byte in the last position of the exchange area. In this way, as the Control Field changes from message to message, the whole message is consistent as long as the first and last bytes are matching.



From the practical point of view, a complete message in the Input area could be considered consistent as soon as the PLC verifies that IN[0] is equal to IN[InputAreaSize - 1].

The PLC should take care to double OUT[0] in OUT[OutputAreaSize - 1] to let the driver check the consistency.

Due to the new byte used by the DAD driver, barcode messages longer than (InputAreaSize – 4) will be split into pieces through an automatic fragmentation process.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Let's analyse a fragmented and consistent data exchange based on:

- Flow Control = **DAD Driver**
- Data Consistency = **Enable**
- Input Area Size = **16**
- Output Area Size = **8**

- After power up Input and Output areas are generally filled by zero. According to DAD driver implementation, Input area has Control Field = IN[0] = **IN[15]** = 80_{Hex} and SAP = 00_{Hex}.

Input Area	**80**$_{Hex}$	**00**$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	**80**$_{Hex}$
Output Area	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$							

- PLC must set Control Field = OUT[0] = **OUT[7]** = 80_{Hex} according to DAD messaging protocol with Data Consistency.

Input Area	80$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	80$_{Hex}$
Output Area	**80**$_{Hex}$	**00**$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	**80**$_{Hex}$							

- DS6X00 reads a barcode with content "*1234567890abcde1234567890abcde*". Let's assume standard data formatting with <STX> as header and <CR><LF> as terminators. DS6X00 transmits first fragment "<STX>*1234567890a*" only (setting More Bit = 1) then it toggles bit A.

Input Area	**89**$_{Hex}$	00$_{Hex}$	**0C**$_{Hex}$	**02**$_{Hex}$	**31**$_{Hex}$	**32**$_{Hex}$	**33**$_{Hex}$	**34**$_{Hex}$	**35**$_{Hex}$	**36**$_{Hex}$	**37**$_{Hex}$	**38**$_{Hex}$	**39**$_{Hex}$	**30**$_{Hex}$	**61**$_{Hex}$	**89**$_{Hex}$
Output Area	80$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	80$_{Hex}$								

- PLC detects transition of bit A so now it can read first incoming fragment (it copies **12** bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	89$_{Hex}$	00$_{Hex}$	0C$_{Hex}$	02$_{Hex}$	31$_{Hex}$	32$_{Hex}$	33$_{Hex}$	34$_{Hex}$	35$_{Hex}$	36$_{Hex}$	37$_{Hex}$	38$_{Hex}$	39$_{Hex}$	30$_{Hex}$	61$_{Hex}$	89$_{Hex}$
Output Area	**81**$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	**81**$_{Hex}$								

- DS6X00 detects transition of bit B so it sends second fragment "*bcde12345678*" (still More Bit = 1) and toggles bit A.

Input Area	**88**$_{Hex}$	00$_{Hex}$	**0C**$_{Hex}$	**62**$_{Hex}$	**63**$_{Hex}$	**64**$_{Hex}$	**65**$_{Hex}$	**31**$_{Hex}$	**32**$_{Hex}$	**33**$_{Hex}$	**34**$_{Hex}$	**35**$_{Hex}$	**36**$_{Hex}$	**37**$_{Hex}$	**38**$_{Hex}$	**88**$_{Hex}$
Output Area	81$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	81$_{Hex}$								

- PLC reads second fragment (it copies **12** bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	88$_{Hex}$	00$_{Hex}$	0C$_{Hex}$	62$_{Hex}$	63$_{Hex}$	64$_{Hex}$	65$_{Hex}$	31$_{Hex}$	32$_{Hex}$	33$_{Hex}$	34$_{Hex}$	35$_{Hex}$	36$_{Hex}$	37$_{Hex}$	38$_{Hex}$	88$_{Hex}$
Output Area	**80**$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	**80**$_{Hex}$								

- DS6X00 sends last fragment "*90abcde<CR><LF>*" (finally More Bit = 0) and toggles bit A.

Input Area	**81**$_{Hex}$	00$_{Hex}$	**09**$_{Hex}$	**39**$_{Hex}$	**30**$_{Hex}$	**61**$_{Hex}$	**62**$_{Hex}$	**63**$_{Hex}$	**64**$_{Hex}$	**65**$_{Hex}$	**0D**$_{Hex}$	**0A**$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	**81**$_{Hex}$
Output Area	80$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	80$_{Hex}$								

- PLC reads last fragment (it copies **9** bytes in its memory from IN[3] on) and now the reassembling can be completed. Then it toggles bit B as acknowledge.

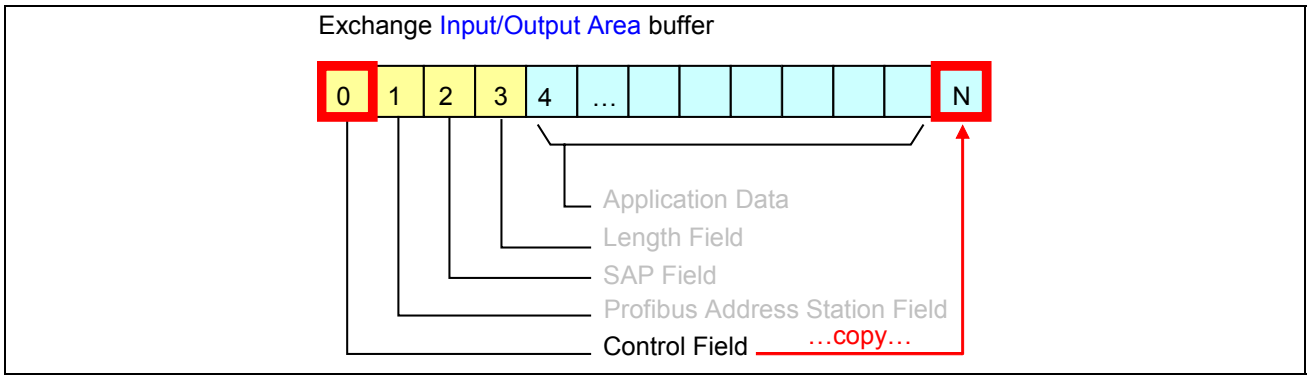
Input Area	81$_{Hex}$	00$_{Hex}$	09$_{Hex}$	39$_{Hex}$	30$_{Hex}$	61$_{Hex}$	62$_{Hex}$	63$_{Hex}$	64$_{Hex}$	65$_{Hex}$	0D$_{Hex}$	0A$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	81$_{Hex}$
Output Area	**81**$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	00$_{Hex}$	**81**$_{Hex}$								

- Whole message has been completely transmitted.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

2.4.2 Data Consistency with DPD Driver

Here the application layer still copies the Control Field byte in the last position of the exchange area. Data exchange structure appears slightly different, as shown in following picture:



If the DPD driver is used, barcode messages longer than (InputAreaSize – **5**) will be split into pieces through an automatic fragmentation process.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

3. Digital I/O Conditioning

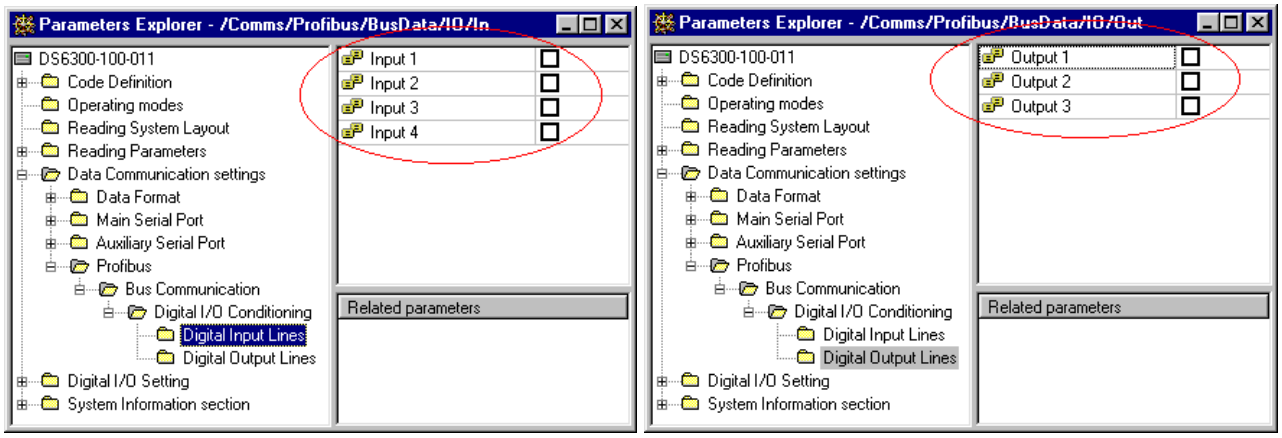
This function offers the possibility to use the scanner I/O as remote peripherals of the Host:

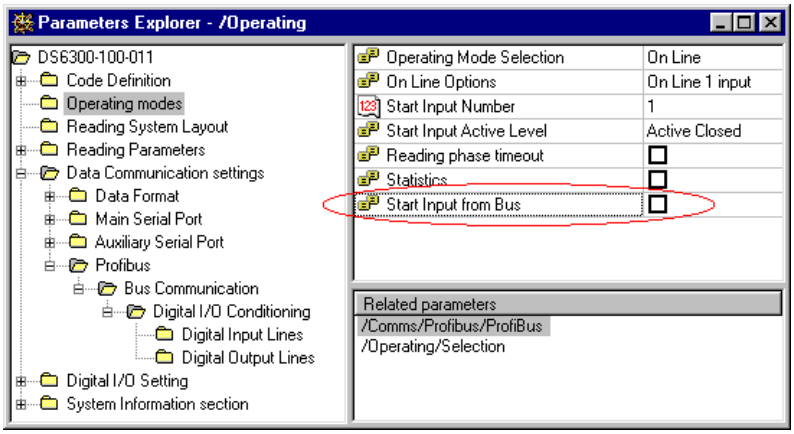
- Digital Output Conditioning allows the PLC to force the level of the three physical DS6X00 outputs
- Digital Input Conditioning allows the DS6X00 to notify the status of its four physical inputs to the PLC

Some of the provided benefits are:

- Profibus Master knows the status of each single scanner input, i.e a parcel is coming, an oversized piece of baggage has been detected or a piece of baggage has reached the data Tx line and so on…
- Profibus Master is able to force the scanner in reading mode
- Profibus Master is able to modify the scanner outputs status depending on the received barcode message (sorting application)
- No dedicated sw on the scanner is necessary to manage its I/Os.

DS6X00 digital I/O conditioning is activated whenever one of the below parameters is checked:





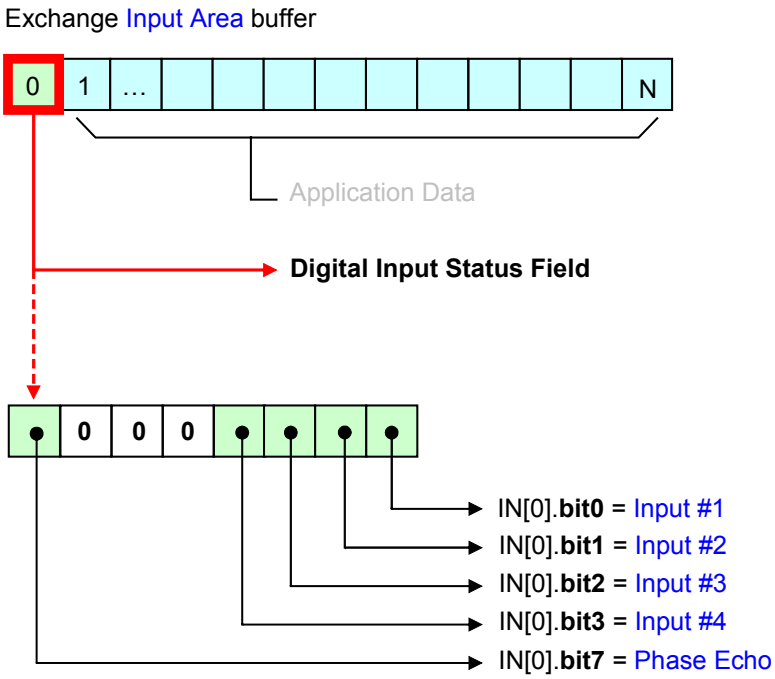
DS6X00 digital I/Os are thus mapped over the first byte of both Input and Output exchange areas, as depicted in the next paragraphs.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

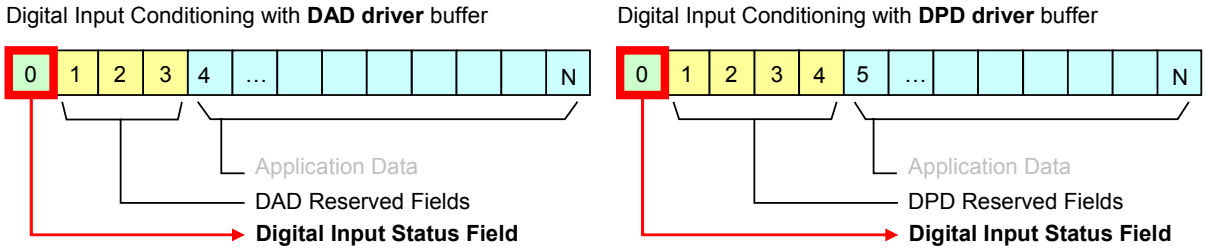
3.1 Digital Input Conditioning

Note: The I/O Conditioning function works independently from the Flow Control options.

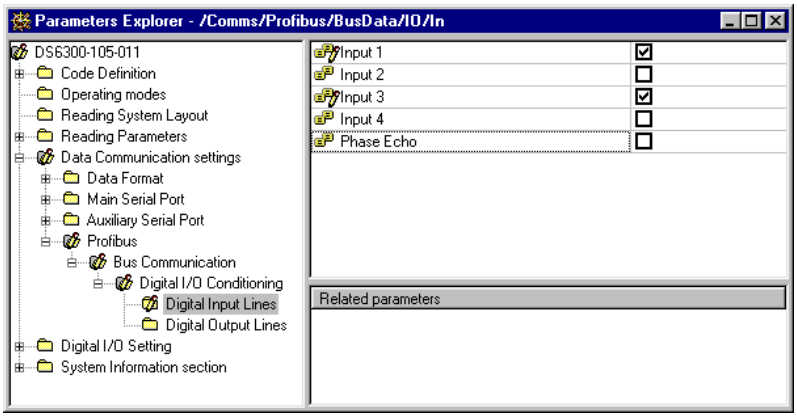
Without any Flow Control driver, the following Input mapping is available:



With Flow Control drivers, the Input area structure is as follows:


Digital Input Conditioning with DAD driver buffer
0 1 2 3 4 … N
Application Data
DAD Reserved Fields
Digital Input Status Field
Digital Input Conditioning with DPD driver buffer
0 1 2 3 4 5 … N
Application Data
DPD Reserved Fields
Digital Input Status Field

The Digital Input Status Field is continuously updated by the DS6X00 according to the status of the physical inputs of the scanner. Only the selected inputs are mirrored, like Input#1 and Input#3 in the figure below:

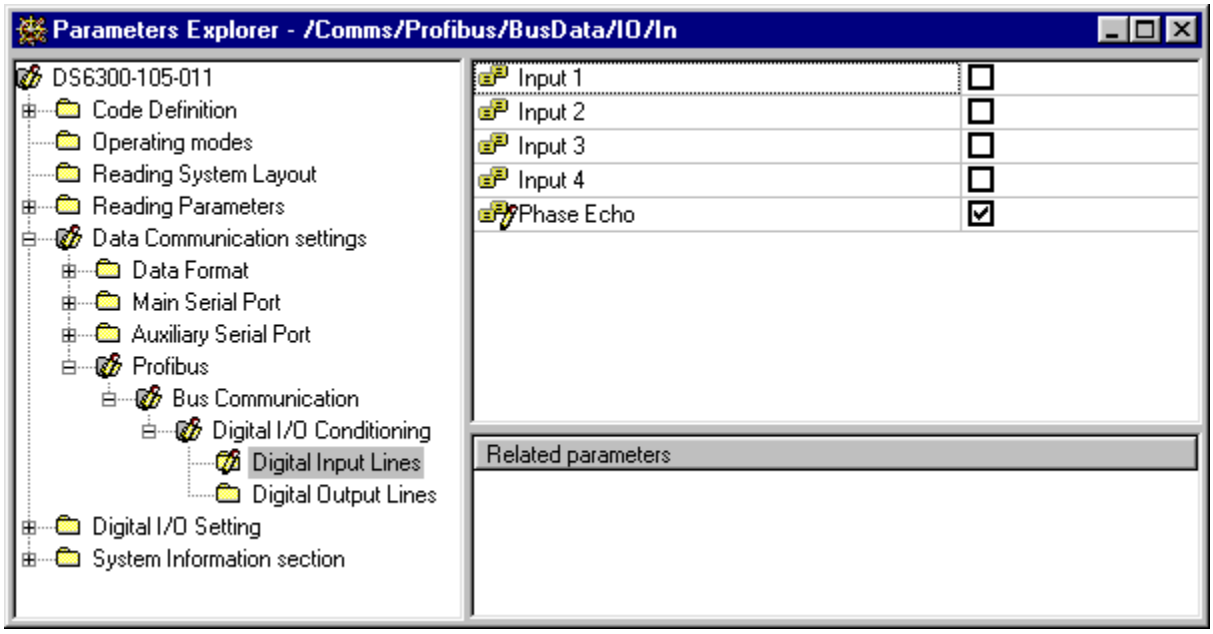


This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

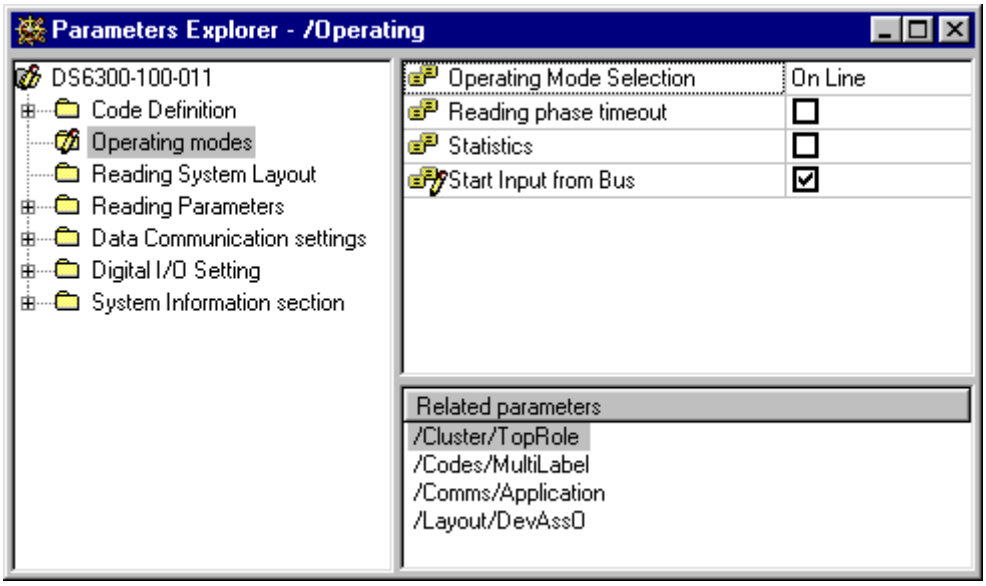
3.1.1 Phase Echo

The Phase Echo bit indicates the status of the reading phase remotely activated on the DS6X00.

This function allows the Master to obtain feedback about the current reading activation via the Profibus network.



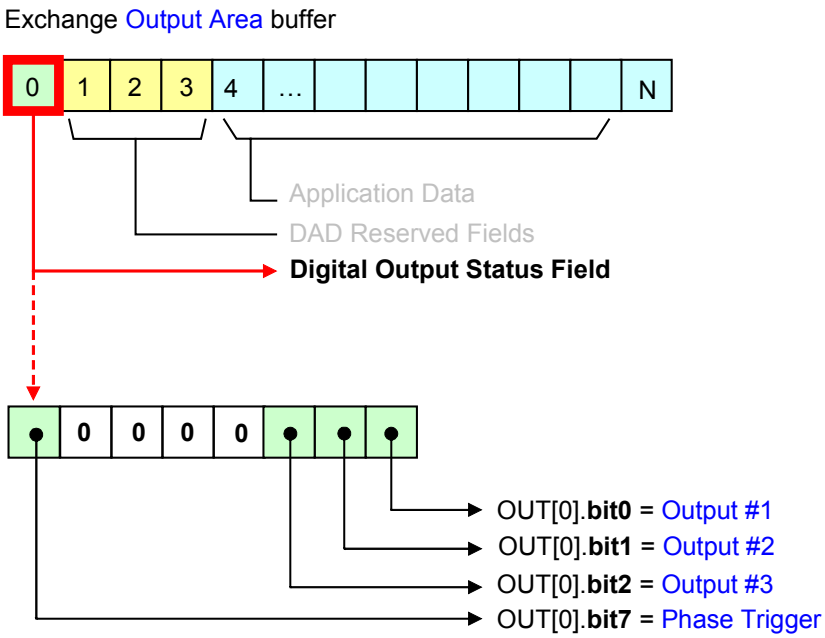
The Phase Echo parameter can be enabled once the Start Input From Bus parameter is selected. See details in the "*Reading Phase via Profibus*" paragraph.



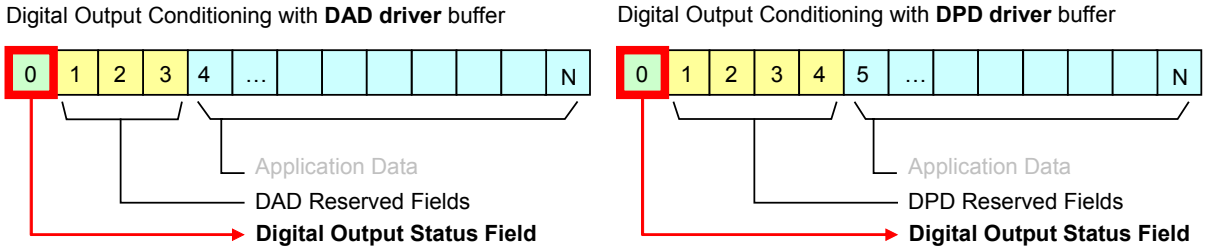
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

3.2 Digital Output Conditioning

Note: The I/O Conditioning function works independently from the Flow Control options.
Without any Flow Control driver, the following Output mapping is available:



With Flow Control drivers, the Output area structure is as follows:



The Digital Output Status Field can be set by the PLC at any time, then the DS6X00 physical outputs will change accordingly. Only the selected outputs are thus controlled.

3.2.1 Phase Trigger

The Phase Trigger bit is used by the Master to force the reading phase of the scanner. Reading activation is provided through the Profibus network so there is no need of external photocells or I/0 signals.

Phase Trigger is active once the Start Input From Bus parameter is selected. See details in the "*Reading Phase via Profibus*" paragraph.

Note: an alternative way to open and close each reading phase via Profibus is letting the DS6X00 operate in Serial On Line mode and then send the user programmable Serial Start String and Serial Stop String through the PLC.

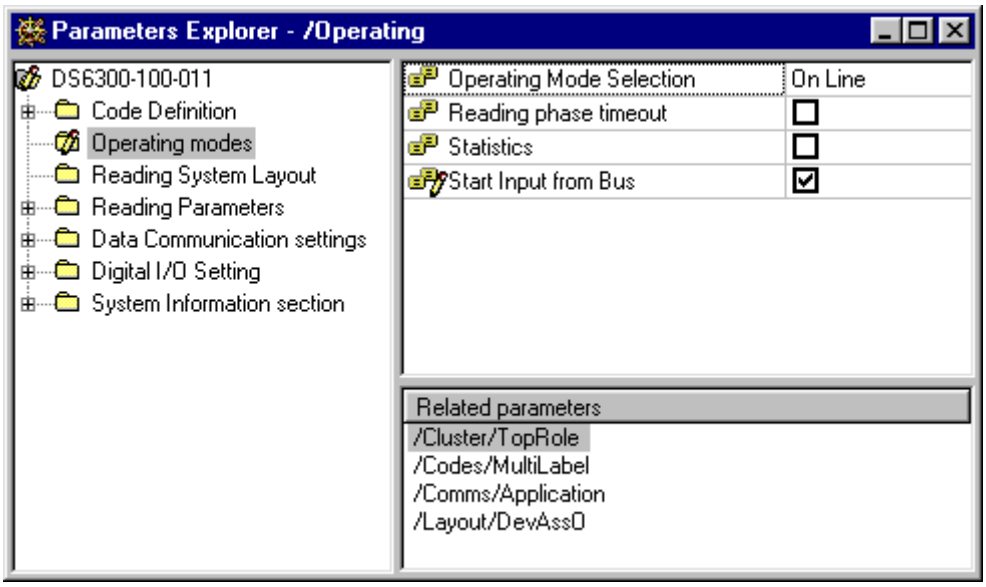
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

3.3 Reading Phase via Profibus

This implementation requires few steps:

1) Connect the scanner with the Genius™ configuration tool

2) Check the Start Input from Bus parameter in the *Operating Modes* folder so the scanner understands that the phase trigger is going to be provided via Profibus



3) Send the modified configuration to the DS6X00 permanent memory (Eeprom)

Now the reading phase of the scanner is controlled as follows:

- Reading phase starts as soon as the Master sets OUT[0].bit7
- Reading phase stops as soon as the Master clears OUT[0].bit7

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

4. Network Configuration

4.1 GSD file

A GSD file is a readable ASCII text file that contains a complete description of the specific device. Basically GSD includes both general info (i.e. vendor and device name, hw/sw releases) and device specific info (Input and Output area size, communication parameters, scanner setup parameters and so on).

Powerful configuration tools (i.e. Siemens SIMATIC Manager) are available to setup a Profibus network. Based on the GSD files, these allow easy configuration of Profibus networks with devices from different manufacturers.

Note: firstly a GSD file must be installed into the PLC environment in order to let a new device be identified and to work on the Profibus network. Follow the instructions in the "*GSD Installation*" paragraph.

DS6X00 is equipped with the following files:

- DTL_07F0.GSD } Device description file
- DS6000.DIB
- DS6000DI.DIB } Custom identification icons
- DS6000SF.DIB

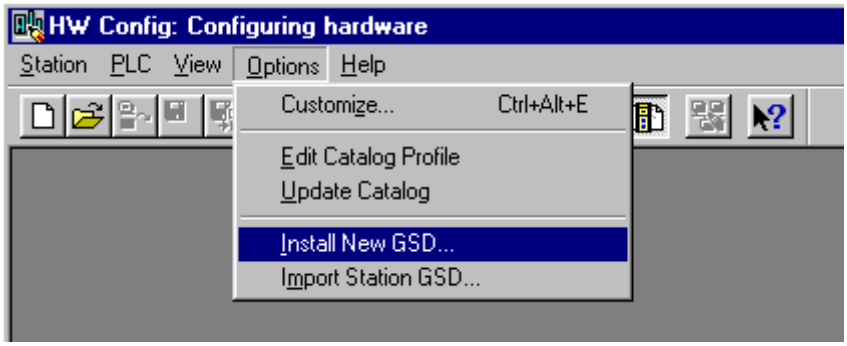
The GSD file is a certified part of the device and must not be changed manually. This file is also not changed by the configuration tool.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

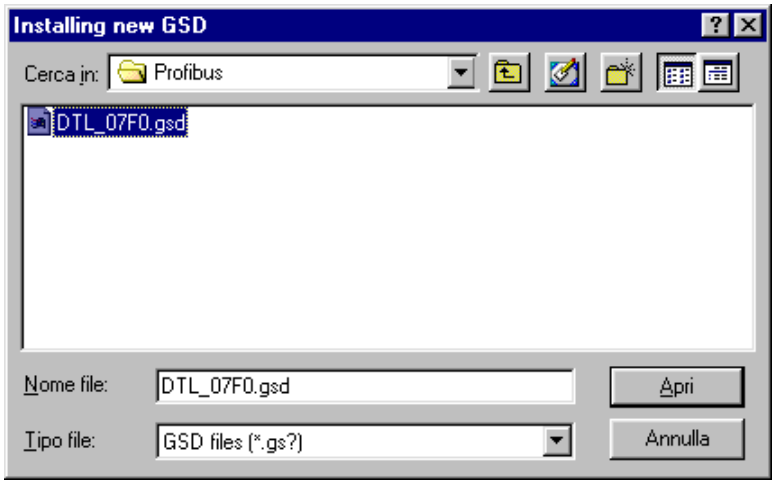
4.2 GSD Installation

The first action that must be done is to add the DS6X00 as a new Profibus-DP Slave among the catalogue of suitable devices for the PLC.

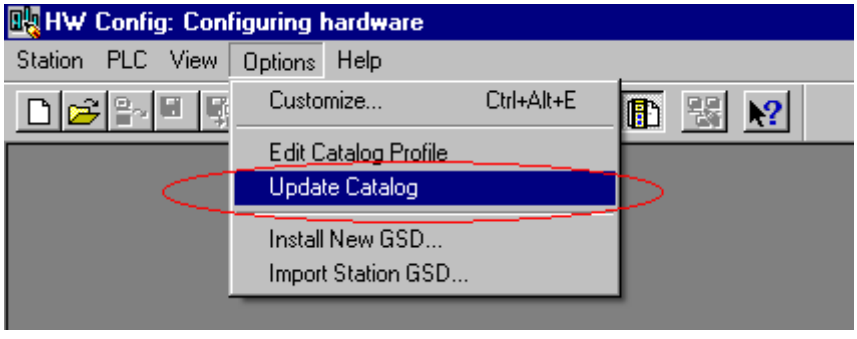
1. *Install the new GSD file...*



2. *Find the GSD file…*
   (GSD and *.DIB files must be in the same directory)

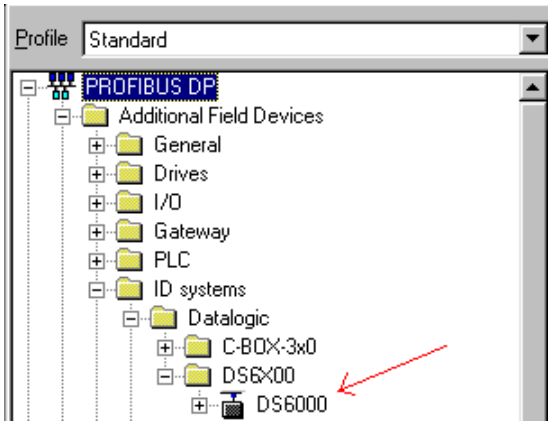


3. *Update Catalogue…*

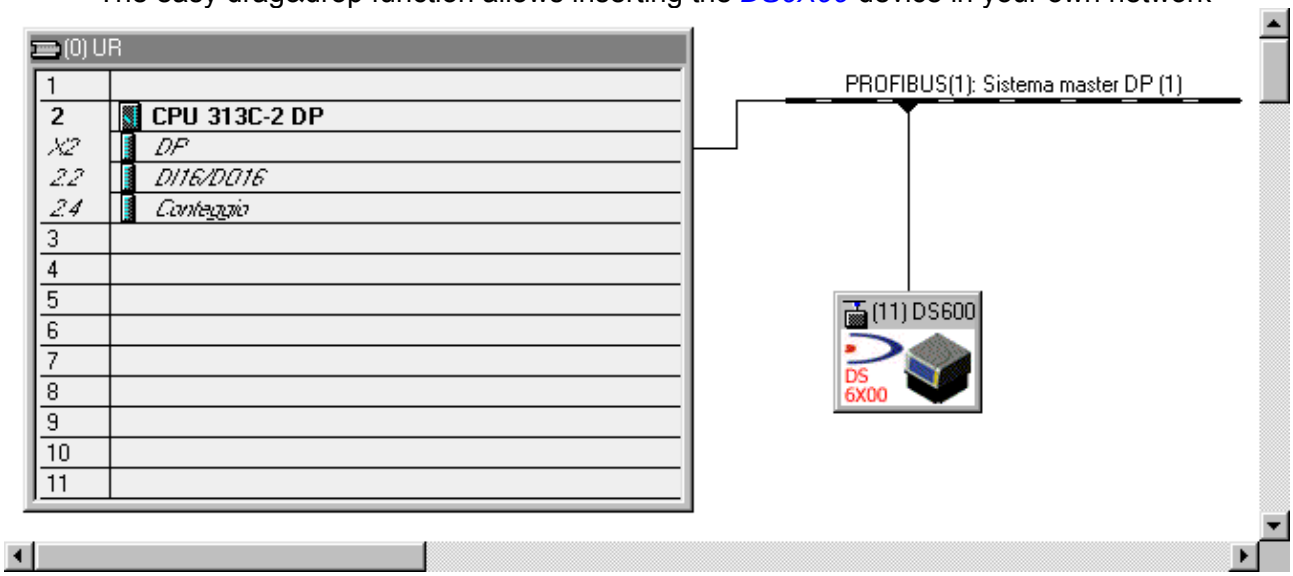


This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

4. *Find new device…*
   A new *DS6000* device will appear in the PLC catalogue under
   Profibus-DP → Additional Field Devices → ID Systems → Datalogic → DS6X00 folder



5. *Insert the device in the Profibus network…*
   The easy drag&drop function allows inserting the *DS6X00* device in your own network



This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

4.3 Scanner Programming via GSD File

The major benefit of the GSD file is to setup the scanner by means of the Profibus Master without the using the Genius™ program: it is possible to select the barcodes, the scanner operating mode and many other relevant parameters.

Different scanner parameters are grouped in Project Modules that can be used to create custom configurations. The User should include the desired modules in his own PLC project and select the proper values for each parameter. Once completed, the PLC will automatically program the scanner at each power on and also any time it is re-discovered on the network (reconnected).

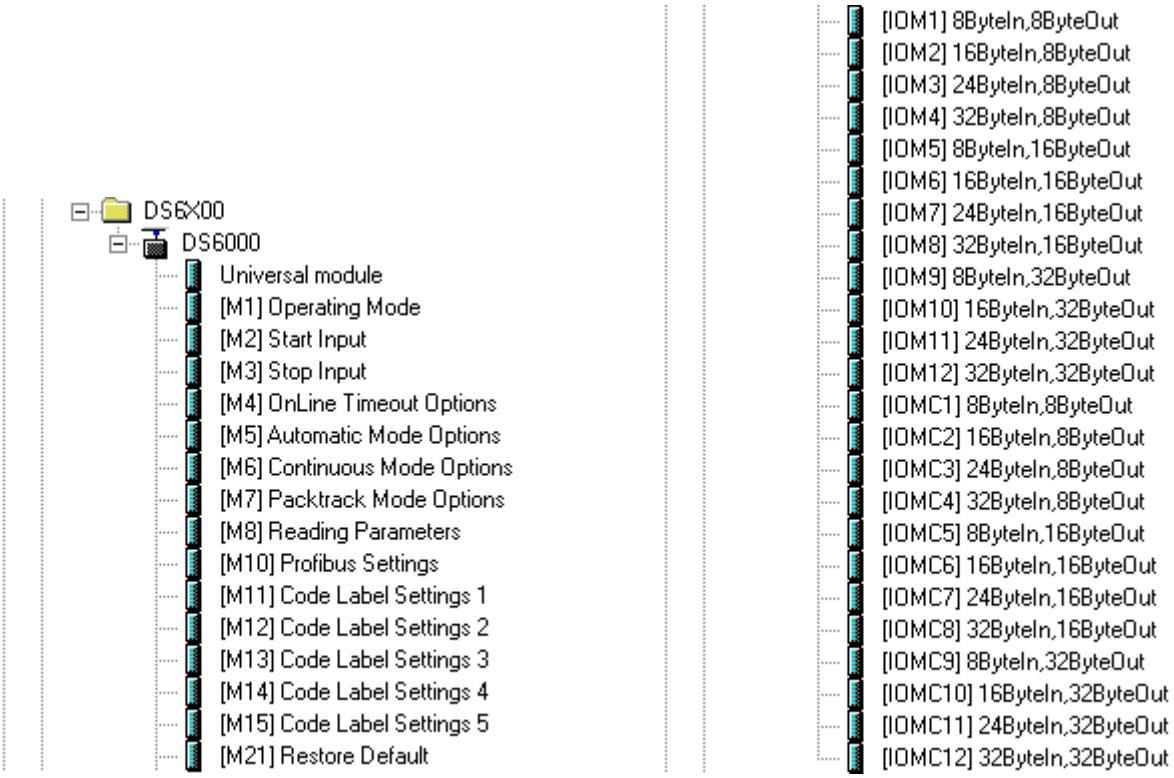
Since the scanner configuration is stored in the PLC, scanner replacement operations become extremely easy and quick.

4.3.1 Project Modules

Three types of modules are available:

- **[IOMCxx]** modules → I/O module with Data Consistency over the whole string
- **[IOMxx]** modules → I/O module with Data Consistency over bytes/words
- **[Mxx]** modules → Scanner Parameters module

Modules overview:



This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

At least one I/O module is required to set the dimensions of both the Input and Output exchange areas. If needed, one or more [**IOMCxx**] (or [**IOMxx**]) modules can be combined.

Scanner parameterisation is optional. In this case, the [M1] Operating Mode module must always be present.

It is suggested to use the [M21] Restore Default module first, to allow a congruent configuration. All of the available modules, except the I/O modules, are generally executed in the same order in which they are inserted.

Let's see the details of each module:

[M1] Operating Mode module

Parameter	*Options*
Operating Mode	"On Line 1In" "On Line 2In" "On Line 1In/Timeout" "On Line ProfiBus" "On Line ProfiBus/Timeout" "Serial On Line" "Serial On Line/Timeout" "Continuous" "Test" "Packtrack" "Automatic"

[M2] Start Input module
[M3] Stop Input module

Parameter	*Options*
Presence Sensor	"Input 1" "Input 2" "Input 3" "Input 4"
PS Active Level	"Active Closed" "Active Open"

[M4] OnLine Timeout Options module

Parameter	*Options*
Timeout	"40(ms)" "50(ms)" "70(ms)" "100(ms)" "150(ms)" "200(ms)" "500(ms)" "1000(ms)" "2000(ms)" "3000(ms)" "4000(ms)" "5000(ms)" "7000(ms)" "10000(ms)" "12000(ms)" "15000(ms)"

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

[M5] Automatic Mode Options module
[M6] Continuous Mode Options module

Parameter	*Options*
Code Filter Depth	"0" "1" "2" "3" "4" "5" "6" "7" "8" "9" "10" "15" "20" "25" "30" "35" "40" "45" "50"
Threshold	"10 scans" "20 scans" "50 scans" "80 scans" "100 scans" "150 scans" "200 scans" "250 scans" "300 scans" "350 scans" "400 scans" "500 scans" "750 scans" "1000 scans" "5000 scans" "10000 scans" "15000 scans" "20000 scans" "30000 scans"

[M7] Packtrack Mode Options module

Parameter	*Options*
Physical Encoder	"Enable" "Disable"
Tx Edge	"Trailing" "Leading"
Max Number Of Packs	"2" … "15"

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

[M8] Reading Parameters module

Parameter	*Options*
Reading Mode	"Reconstruction" "Linear"
Overflow Ratio	"0" … "50"
Beam Shutter	"Disable" "Triggered" "Enable"

[M10] Profibus Settings module

Parameter	*Options*
Flow Control	"Disable" "DAD Driver" "DPD Driver"
Data Consistency	"Enable" "Disable"
Input 1	"Enable" "Disable"
Input 2	"Enable" "Disable"
Input 3	"Enable" "Disable"
Input 4	"Enable" "Disable"
Phase Echo	"Enable" "Disable"
Output 1	"Enable" "Disable"
Output 2	"Enable" "Disable"
Output 3	"Enable" "Disable"

[M11] Code Label Setting 1 … [M15] Code Label Setting 5 modules

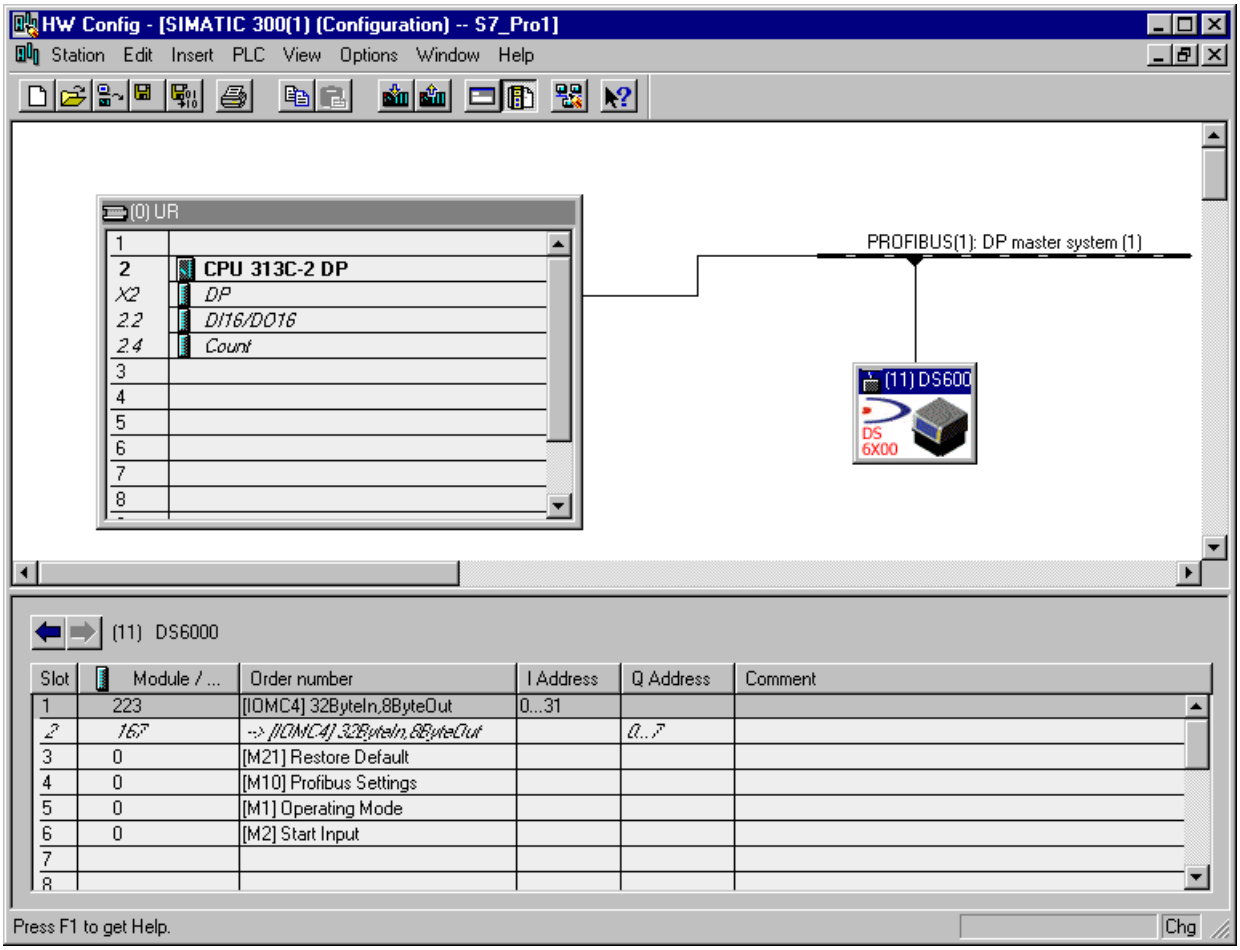
Parameter	*Options*
Code Type	"Code 128" "Interleaved 2 of 5" "Code 39" "Code EAN 128" "EAN-13" "EAN-8" "UPC-A" "UPC-E" "All EAN_UPC" "CODABAR" "Code 93"
Label Length	"Variable" "1" … "48"

[M21] Restore Default module

Parameter	*Options*
<No Parameters>	<No options>

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

A typical scanner programming project is as follows:



Here you have:

[IOMC4] module to set the Input/Output exchange areas.
[M21] module to restore the scanner configuration to Default.
[M10] module to define possible Flow Control protocol and/or Data Consistency options.
[M1] module to select the scanner operating mode, i.e. OnLine mode.
[M2] module to indicate which digital Input provides the trigger.

See also details of the demo project in the following chapter.

Complete parameter descriptions are available in the Genius™ On Line Help.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

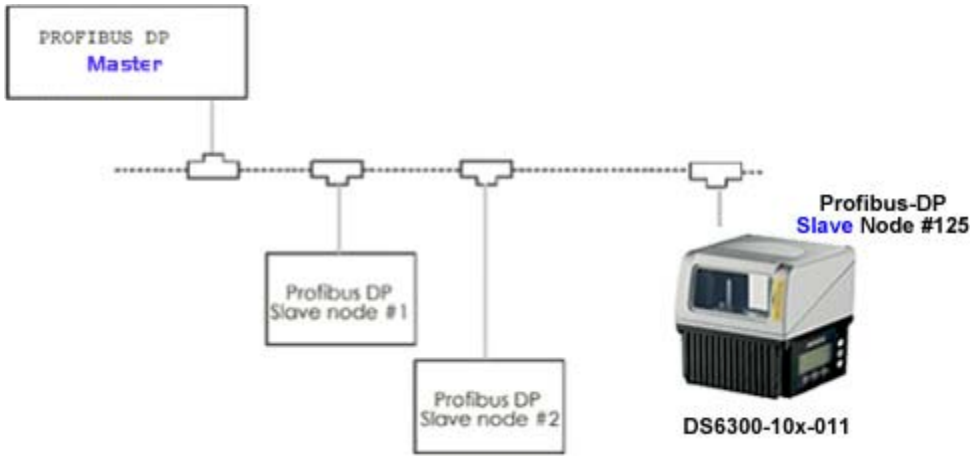
5. Example Application

This chapter describes how to connect a DS6X00-10X-011 barcode reader to a network Mastered by a Siemens S7 PLC.

It is intended to be used along with its accompanying S7 project, scanner configurations and GSD file as a demonstration of Profibus connectivity.

The Application program implements complete data collection by using the DPD Driver protocol.

System Layout



DS6X00-10X-011 Settings

Nothing to do. Scanner has to be taken out from the box and just plugged into the network.

Default Profibus address = 125_{DEC}

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

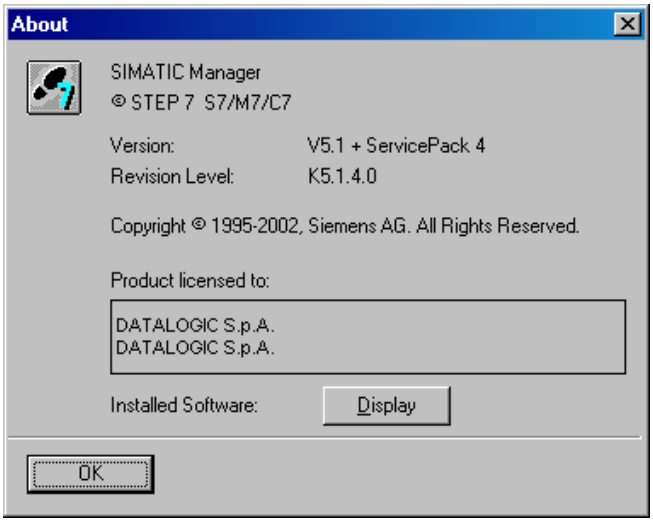
5.1 Installing the Scanner to the Network

Follow the instructions in the "*GSD Installation*" paragraph.

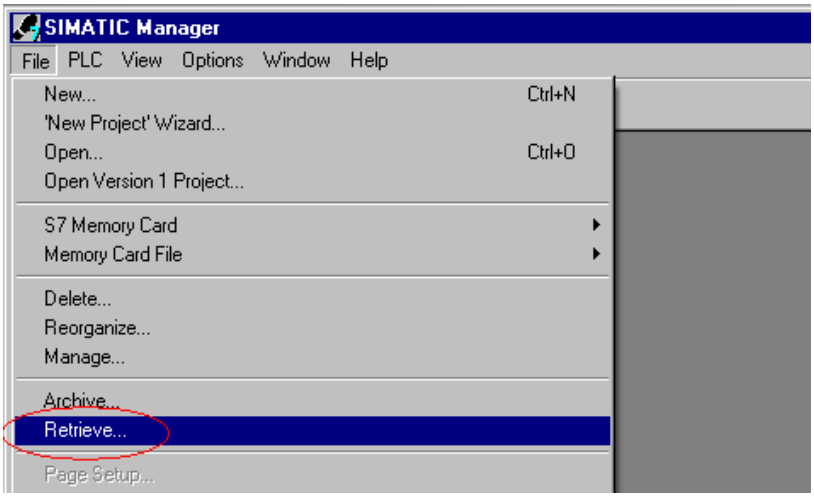
5.2 Loading the Demo Application Program

The Demo Application Program has been designed by means of the Siemens SIMATIC STEP7 V5.1 and has been tested on the Siemens S7 PLC.

1. Run the Siemens SIMATIC STEP7 program.

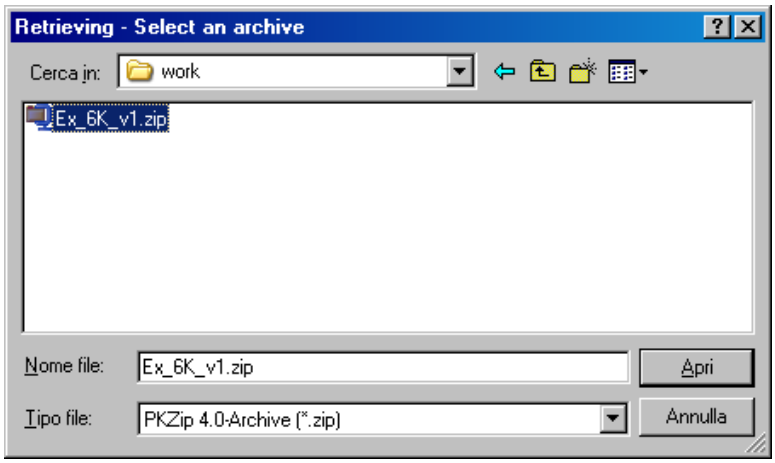


2. *Retrieve…* (Import) the Demo Program.

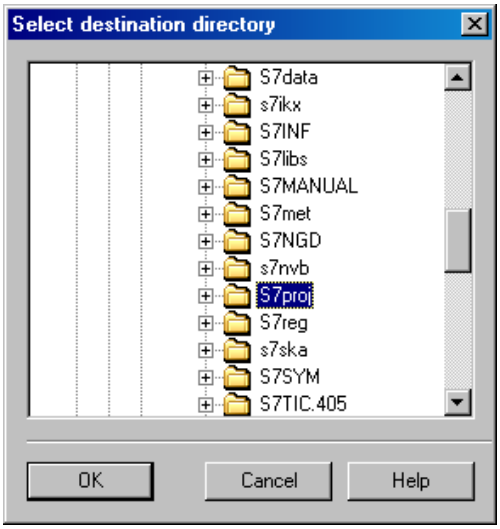


3. *Select Archive…* (attached file is called Ex_6K_v1.zip)

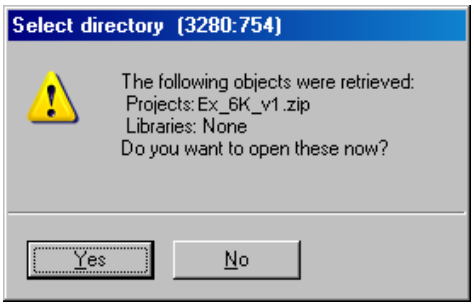
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.



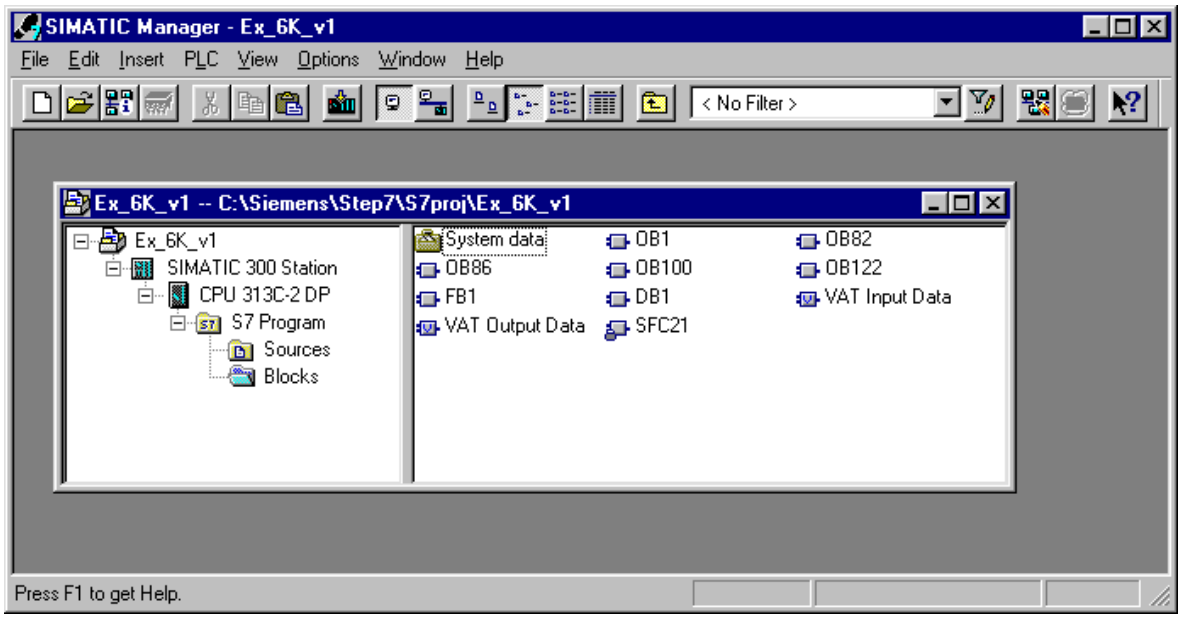
We suggest using the default *Destination Directory* (S7proj)



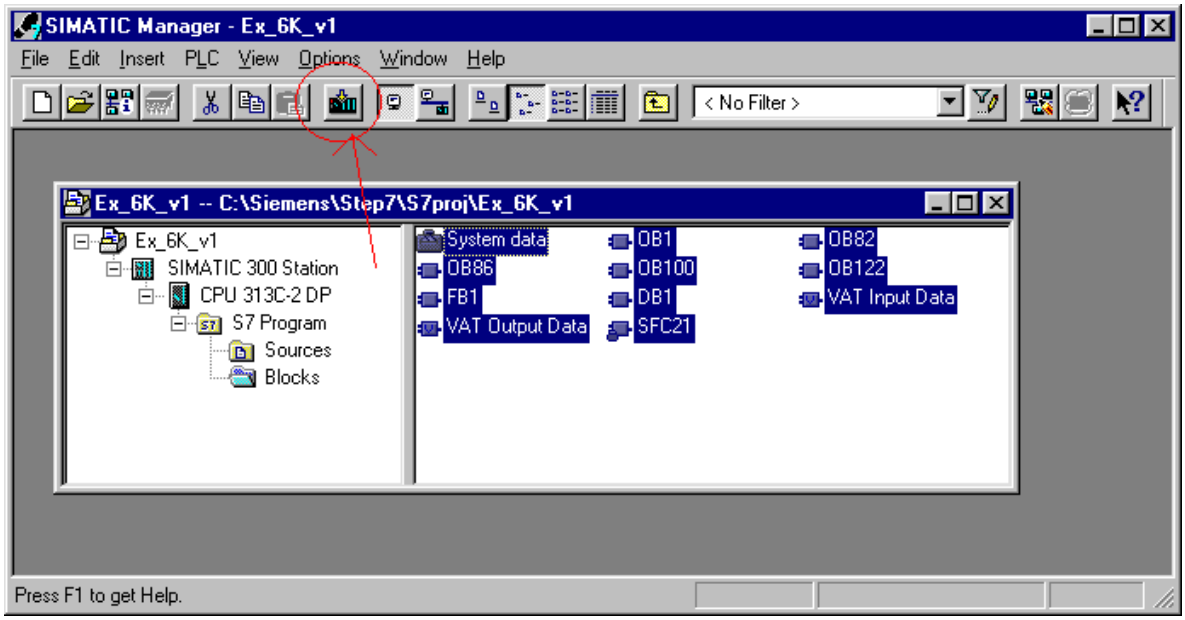
4. Open the project. After unzipping (click Yes to confirm).



This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.



5. Transfer Demo Application Program to the PLC
Select all the Organising Blocks then press the *Download* icon to transfer sw & hw configuration to your PLC.



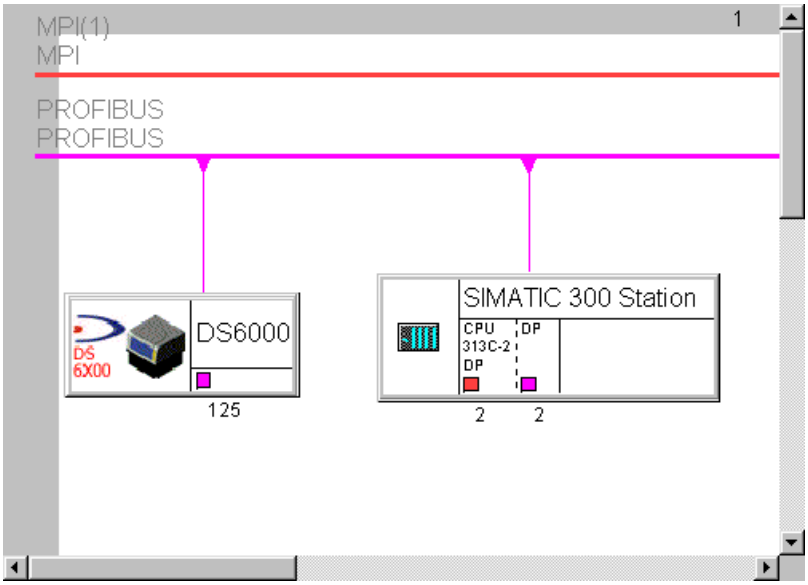
Note: the Demo Application Program was tested with CPU 313-2-DP. In order to avoid possible incompatibilities, the 300 series CPU is recommended.

In any case, the present PLC sw program can be easily adapted to your actual hw/sw configurations.

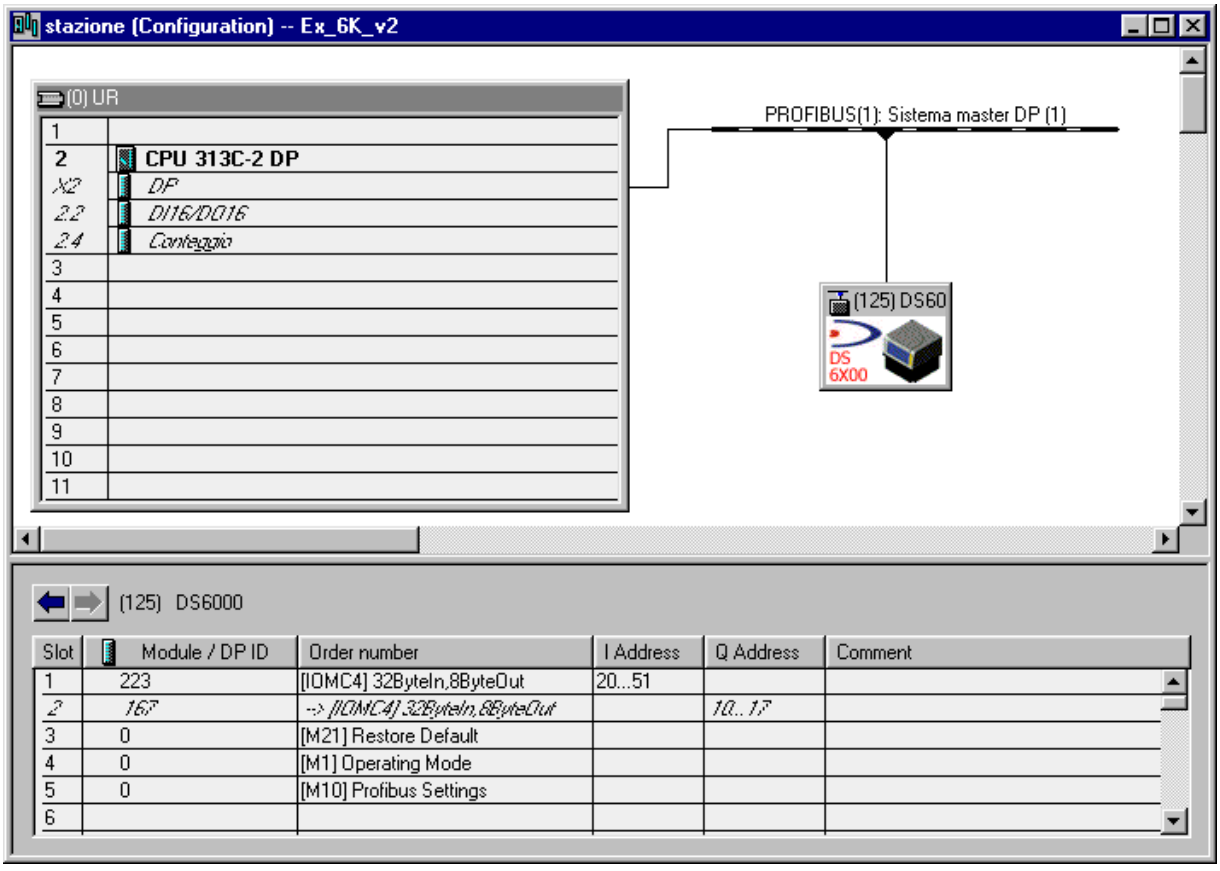
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

5.3 Network Configuration Description

The basic situation is depicted below:



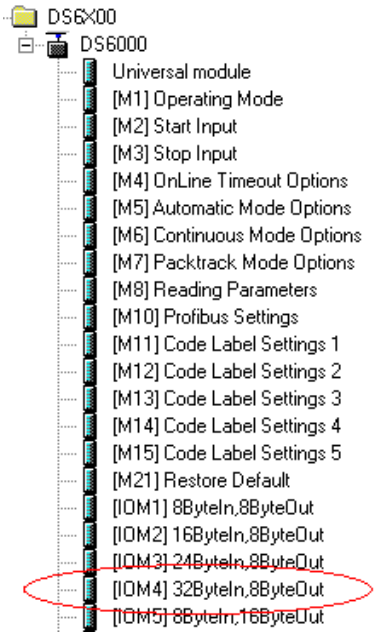
The PLC communicates with the unique DS6X00 set with address 125_{DEC}. No further devices are expected to be present on the Profibus-DP network.



This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

[IOM4] module set the exchange areas as follows:

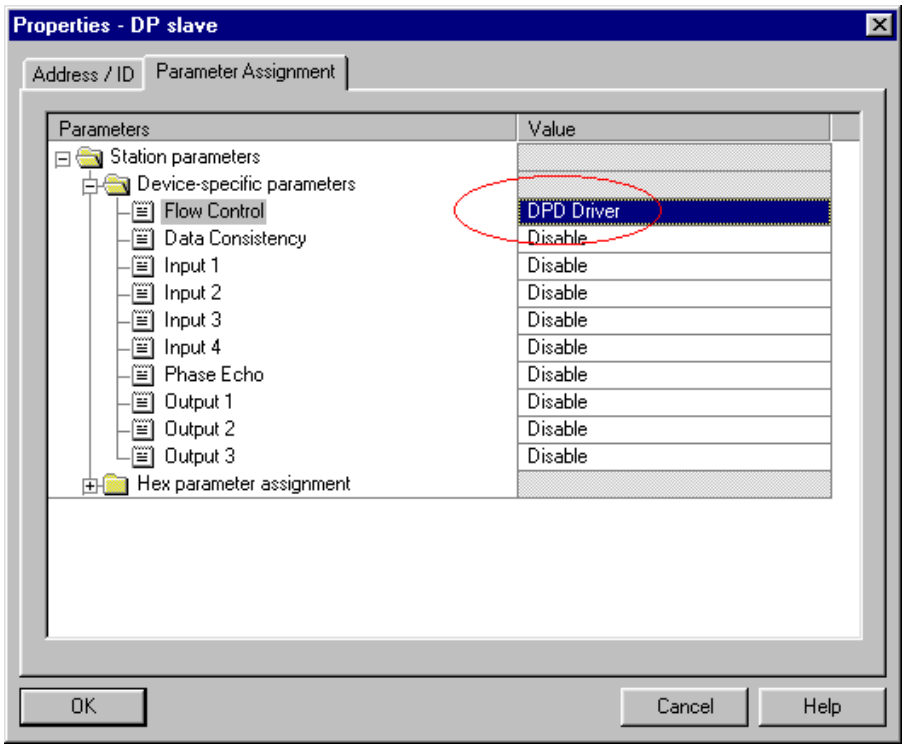
- Input area size = 32 byte
  [mapped from `I20 (E20)` to `I51 (E51)`]
- Output area size = 8 byte
  [mapped from `Q10 (E10)` to `Q17 (E17)`]



[M21] Restore Default assures consistent programming of the scanner even if configuration has been modified previously.

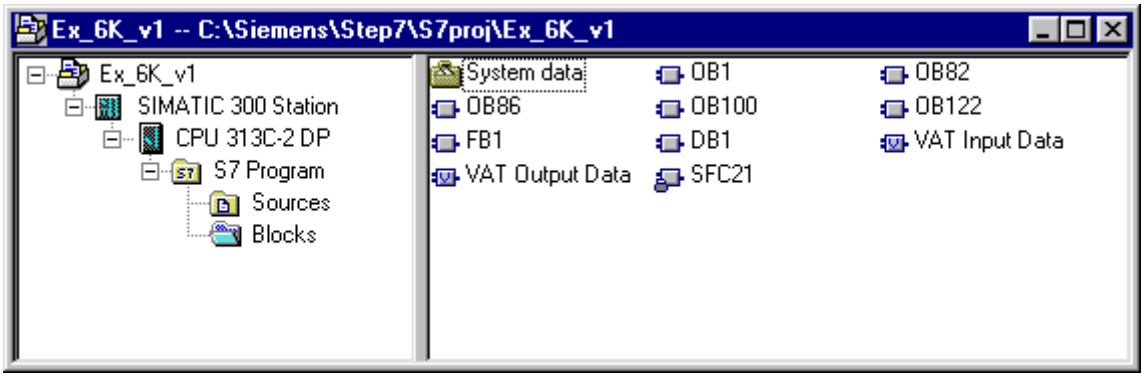
[M1] Operating Mode selects OnLine as the operating mode of the scanner (as default, Input1 is the trigger).

[M10] Profibus Settings enables the Flow Control handshake:



This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

5.3.1 Block Descriptions



The Demo project is made up of the following blocks:

OB1 - Organising Block	"Main Program Sweep (Cycle)"
OB82 - Organising Block	"I/O Point Fault"
OB86 - Organising Block	"Loss Of Rack Fault"
OB100 - Organising Block	"Complete Restart"
OB122 - Organising Block	"Module Access Error"
FB1 - Function Block	"Flow Control Implementation"
DB1 - Data Block	"Data Buffer"
SFC21 - System Function Call	"FILL": initialise a destination memory area with the content of a starting memory area.
VAT Input Data	→ Variable Table to check incoming Data
VAT Output Data	→ Variable Table to check outgoing Data

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

VAT Input Data

→ Variable Table to check incoming Data

VAT Input Data -- Ex_v100\stazione\CPU 313C-2 DP\Programma S7

	Operando	Simbolo	Form	Valore di stato	Valore di comando
1	// Bit 0 of Byte 0 of InputArea: TxBufferEmpty bit				
2	E 20.0		BOOL		
3					
4	// STRUCTURE OF INPUT AREA WITH FLOW CONTROL				
5	// Byte 0 of Input Area: Control Field				
6	EB 20		BIN		
7	// Byte 1 of Input Area: Profibus Station Address Field				
8	EB 21		HEX		
9	// Byte 2 of Input Area: SAP Field				
10	EB 22		HEX		
11	// Byte 3 of Input Area: Length Field				
12	EB 23		HEX		
13	// Byte 4-31 of Input Area: Data Buffer				
14	EB 24		HEX		
15	EB 25		HEX		
16	EB 26		HEX		
17	EB 27		HEX		
18	EB 28		HEX		
19	EB 29		HEX		
20	EB 30		HEX		
21	EB 31		HEX		
22	EB 32		HEX		
23	EB 33		HEX		
24	EB 34		HEX		
25	EB 35		HEX		
26	EB 36		HEX		
27	EB 37		HEX		
28	EB 38		HEX		
29	EB 39		HEX		
30	EB 40		HEX		
31	EB 41		HEX		
32	EB 42		HEX		
33	EB 43		HEX		
34	EB 44		HEX		

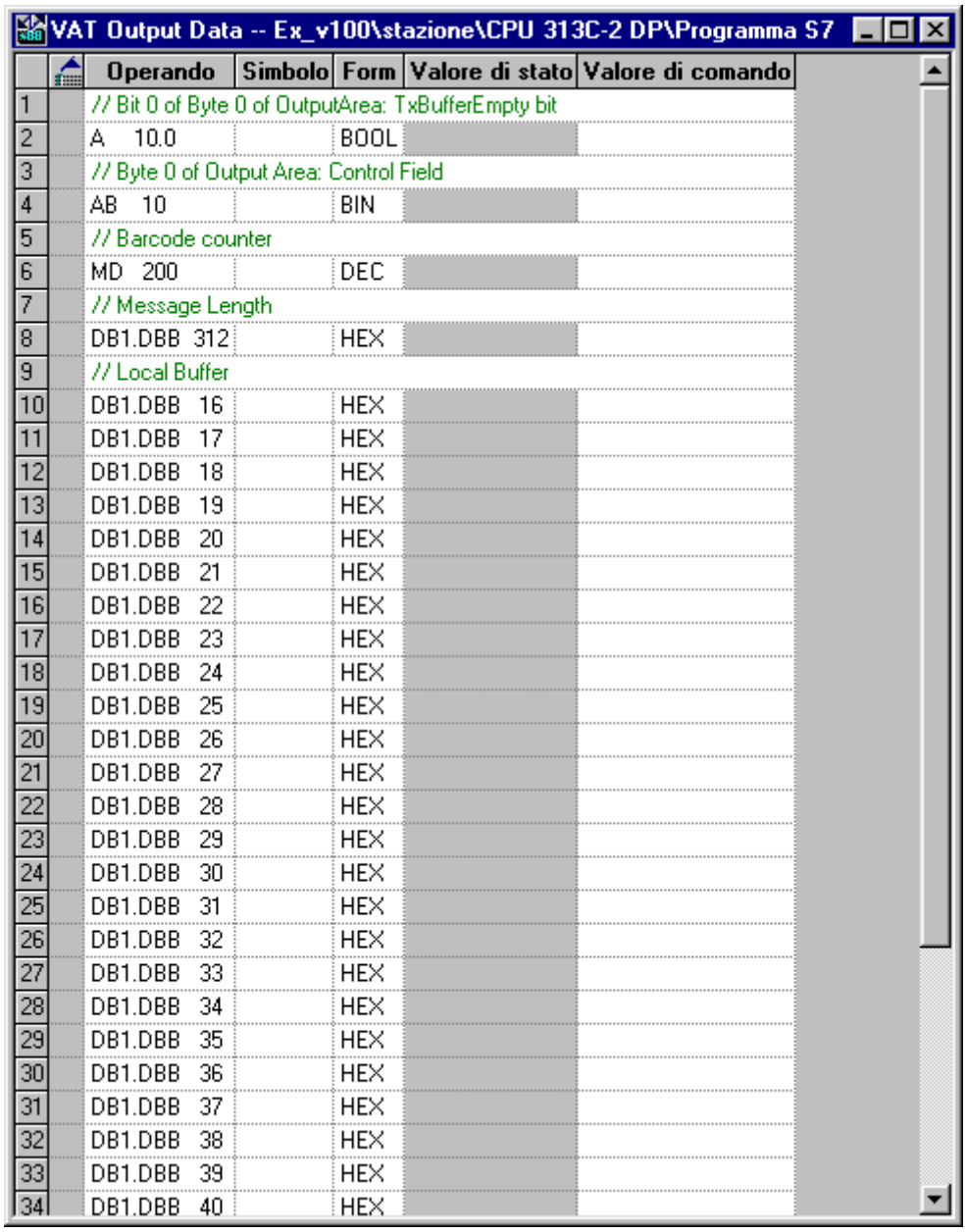
Basically you can check the shared Input area where the scanner writes data to the PLC.

Refer to the "*Data Consistency with DPD Driver*" paragraph for details.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

VAT Output Data

→ Variable Table to check outgoing Data



VAT Output Data -- Ex_v100\stazione\CPU 313C-2 DP\Programma S7

	Operando	Simbolo	Form	Valore di stato	Valore di comando
1	// Bit 0 of Byte 0 of OutputArea: TxBufferEmpty bit				
2	A 10.0		BOOL		
3	// Byte 0 of Output Area: Control Field				
4	AB 10		BIN		
5	// Barcode counter				
6	MD 200		DEC		
7	// Message Length				
8	DB1.DBB 312		HEX		
9	// Local Buffer				
10	DB1.DBB 16		HEX		
11	DB1.DBB 17		HEX		
12	DB1.DBB 18		HEX		
13	DB1.DBB 19		HEX		
14	DB1.DBB 20		HEX		
15	DB1.DBB 21		HEX		
16	DB1.DBB 22		HEX		
17	DB1.DBB 23		HEX		
18	DB1.DBB 24		HEX		
19	DB1.DBB 25		HEX		
20	DB1.DBB 26		HEX		
21	DB1.DBB 27		HEX		
22	DB1.DBB 28		HEX		
23	DB1.DBB 29		HEX		
24	DB1.DBB 30		HEX		
25	DB1.DBB 31		HEX		
26	DB1.DBB 32		HEX		
27	DB1.DBB 33		HEX		
28	DB1.DBB 34		HEX		
29	DB1.DBB 35		HEX		
30	DB1.DBB 36		HEX		
31	DB1.DBB 37		HEX		
32	DB1.DBB 38		HEX		
33	DB1.DBB 39		HEX		
34	DB1.DBB 40		HEX		

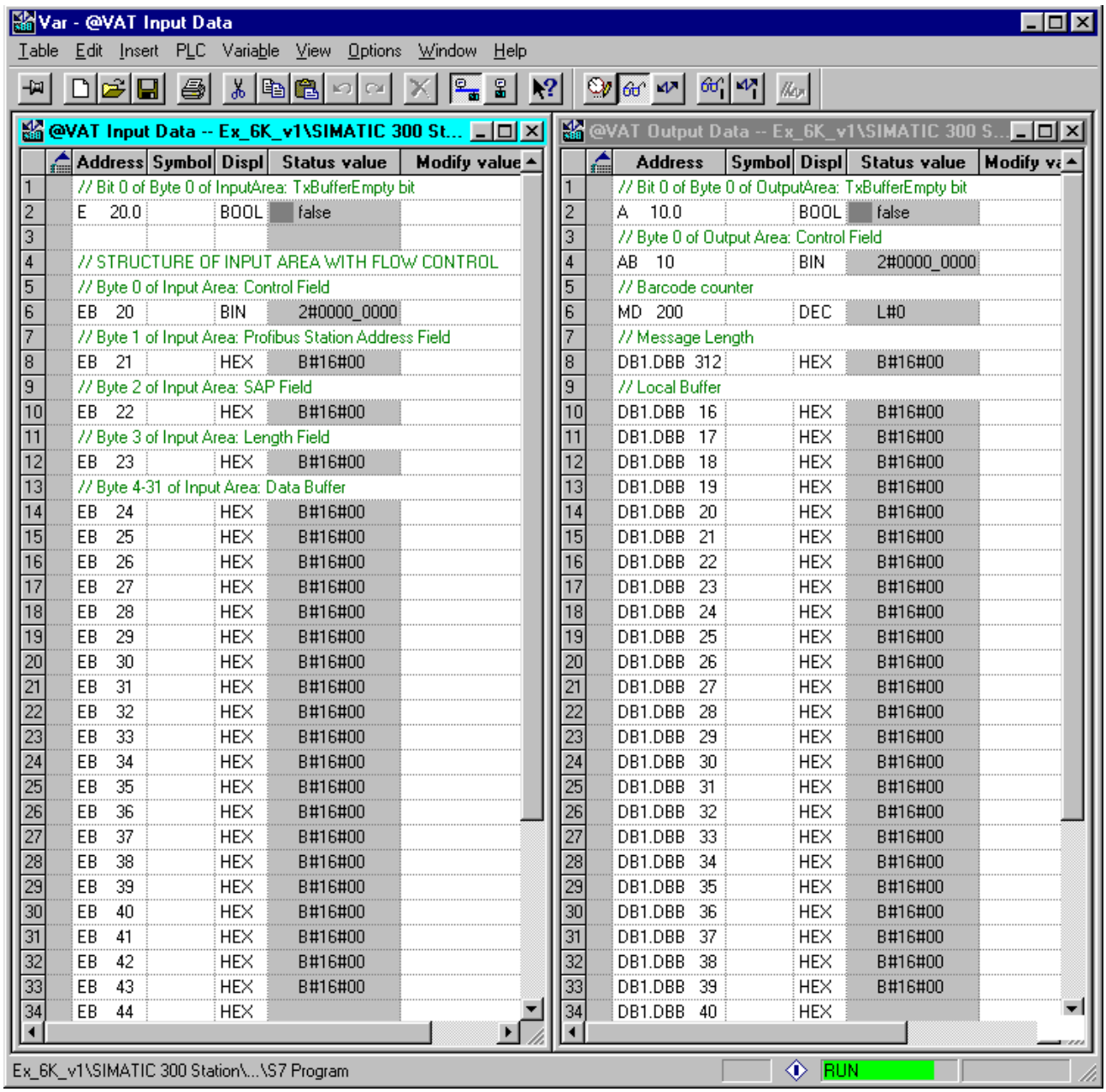
Through this table you will see how many barcodes have been received and the last barcode sent from the reader and moved to the temporary Local Buffer.

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

5.4 Example of Data Exchange

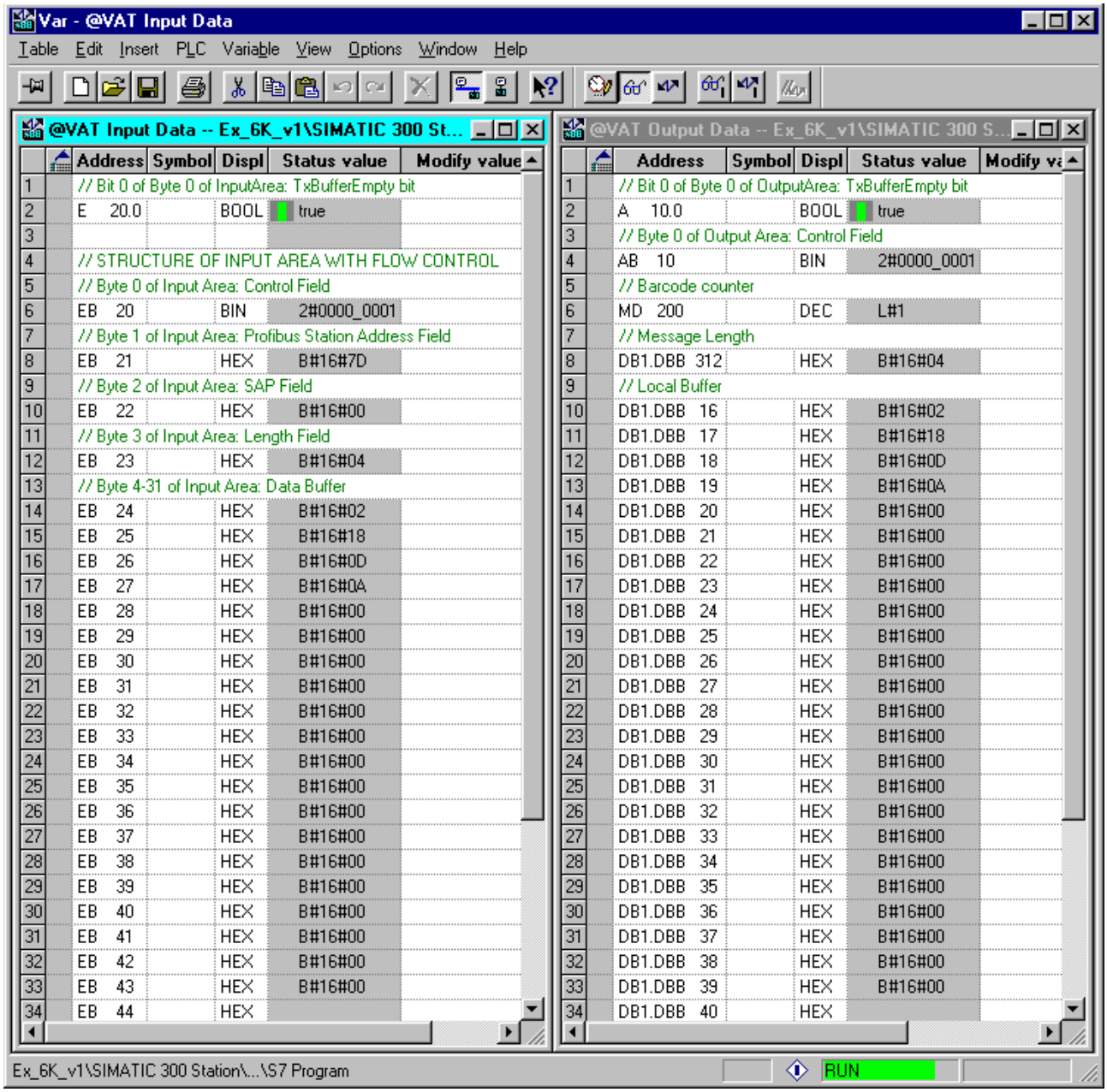
To check the behaviour of Demo Program it is useful switching Step7 to OnLine mode and having a look at the VAT Input Area and the VAT Output Area.

After power on, starting values should be as follows:


Var - @VAT Input Data
Table Edit Insert PLC Variable View Options Window Help
@VAT Input Data -- Ex_6K_v1\SIMATIC 300 St...
Address Symbol Displ Status value Modify value
1 // Bit 0 of Byte 0 of InputArea: TxBufferEmpty bit
2 E 20.0 BOOL false
3
4 // STRUCTURE OF INPUT AREA WITH FLOW CONTROL
5 // Byte 0 of Input Area: Control Field
6 EB 20 BIN 2#0000_0000
7 // Byte 1 of Input Area: Profibus Station Address Field
8 EB 21 HEX B#16#00
9 // Byte 2 of Input Area: SAP Field
10 EB 22 HEX B#16#00
11 // Byte 3 of Input Area: Length Field
12 EB 23 HEX B#16#00
13 // Byte 4-31 of Input Area: Data Buffer
14 EB 24 HEX B#16#00
15 EB 25 HEX B#16#00
16 EB 26 HEX B#16#00
17 EB 27 HEX B#16#00
18 EB 28 HEX B#16#00
19 EB 29 HEX B#16#00
20 EB 30 HEX B#16#00
21 EB 31 HEX B#16#00
22 EB 32 HEX B#16#00
23 EB 33 HEX B#16#00
24 EB 34 HEX B#16#00
25 EB 35 HEX B#16#00
26 EB 36 HEX B#16#00
27 EB 37 HEX B#16#00
28 EB 38 HEX B#16#00
29 EB 39 HEX B#16#00
30 EB 40 HEX B#16#00
31 EB 41 HEX B#16#00
32 EB 42 HEX B#16#00
33 EB 43 HEX B#16#00
34 EB 44 HEX
@VAT Output Data -- Ex_6K_v1\SIMATIC 300 S...
Address Symbol Displ Status value Modify va
1 // Bit 0 of Byte 0 of OutputArea: TxBufferEmpty bit
2 A 10.0 BOOL false
3 // Byte 0 of Output Area: Control Field
4 AB 10 BIN 2#0000_0000
5 // Barcode counter
6 MD 200 DEC L#0
7 // Message Length
8 DB1.DBB 312 HEX B#16#00
9 // Local Buffer
10 DB1.DBB 16 HEX B#16#00
11 DB1.DBB 17 HEX B#16#00
12 DB1.DBB 18 HEX B#16#00
13 DB1.DBB 19 HEX B#16#00
14 DB1.DBB 20 HEX B#16#00
15 DB1.DBB 21 HEX B#16#00
16 DB1.DBB 22 HEX B#16#00
17 DB1.DBB 23 HEX B#16#00
18 DB1.DBB 24 HEX B#16#00
19 DB1.DBB 25 HEX B#16#00
20 DB1.DBB 26 HEX B#16#00
21 DB1.DBB 27 HEX B#16#00
22 DB1.DBB 28 HEX B#16#00
23 DB1.DBB 29 HEX B#16#00
24 DB1.DBB 30 HEX B#16#00
25 DB1.DBB 31 HEX B#16#00
26 DB1.DBB 32 HEX B#16#00
27 DB1.DBB 33 HEX B#16#00
28 DB1.DBB 34 HEX B#16#00
29 DB1.DBB 35 HEX B#16#00
30 DB1.DBB 36 HEX B#16#00
31 DB1.DBB 37 HEX B#16#00
32 DB1.DBB 38 HEX B#16#00
33 DB1.DBB 39 HEX B#16#00
34 DB1.DBB 40 HEX
Ex_6K_v1\SIMATIC 300 Station\...\S7 Program
RUN

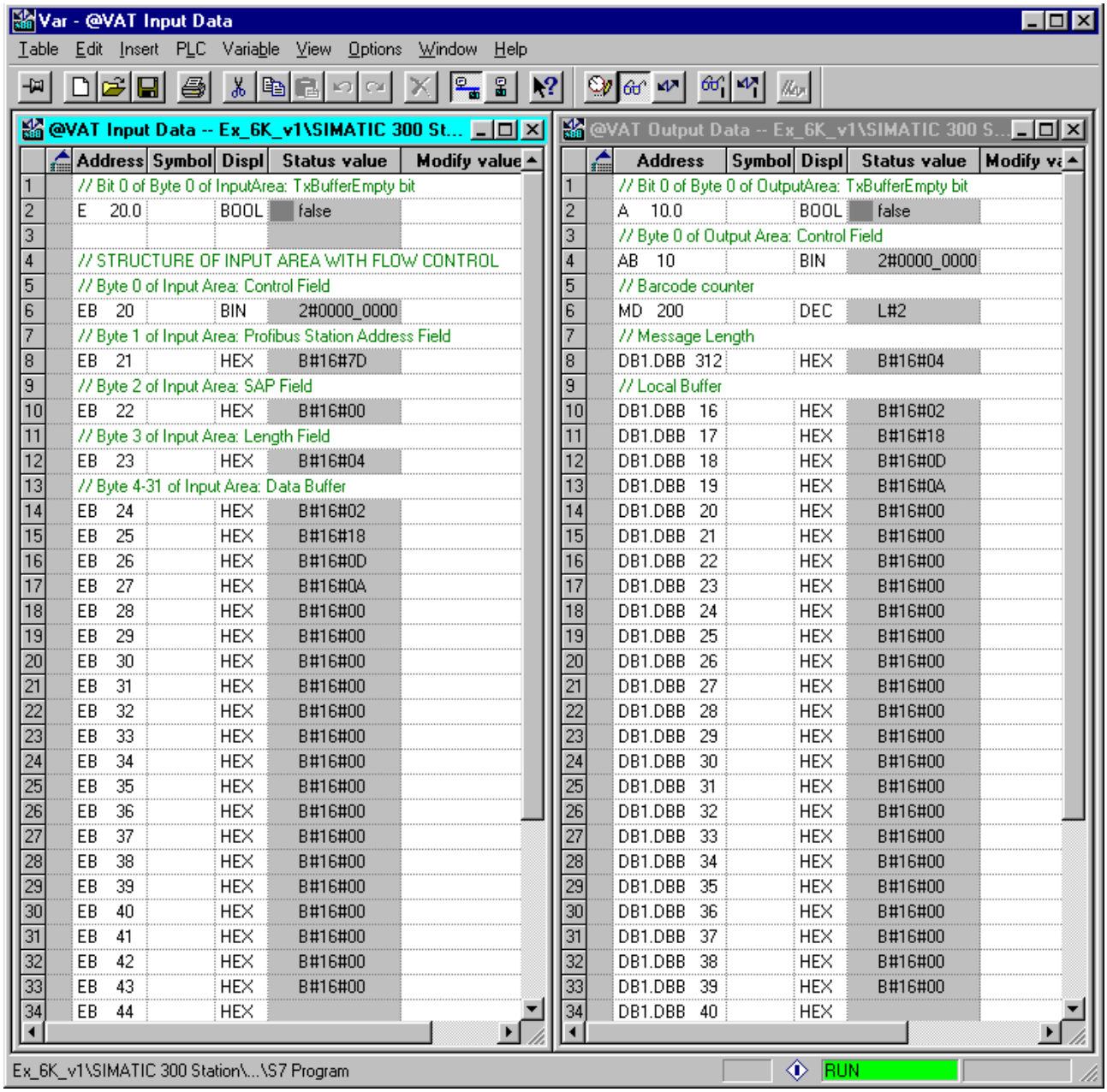
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Let's give a 1st trigger to the scanner making a NoRead.
1st message (NoRead) will be sent to the PLC.
Note Bit 0 toggling.


Var - @VAT Input Data
Table Edit Insert PLC Variable View Options Window Help
@VAT Input Data -- Ex_6K_v1\SIMATIC 300 St...
Address Symbol Displ Status value Modify value
1 // Bit 0 of Byte 0 of InputArea: TxBufferEmpty bit
2 E 20.0 BOOL true
3
4 // STRUCTURE OF INPUT AREA WITH FLOW CONTROL
5 // Byte 0 of Input Area: Control Field
6 EB 20 BIN 2#0000_0001
7 // Byte 1 of Input Area: Profibus Station Address Field
8 EB 21 HEX B#16#7D
9 // Byte 2 of Input Area: SAP Field
10 EB 22 HEX B#16#00
11 // Byte 3 of Input Area: Length Field
12 EB 23 HEX B#16#04
13 // Byte 4-31 of Input Area: Data Buffer
14 EB 24 HEX B#16#02
15 EB 25 HEX B#16#18
16 EB 26 HEX B#16#0D
17 EB 27 HEX B#16#0A
18 EB 28 HEX B#16#00
19 EB 29 HEX B#16#00
20 EB 30 HEX B#16#00
21 EB 31 HEX B#16#00
22 EB 32 HEX B#16#00
23 EB 33 HEX B#16#00
24 EB 34 HEX B#16#00
25 EB 35 HEX B#16#00
26 EB 36 HEX B#16#00
27 EB 37 HEX B#16#00
28 EB 38 HEX B#16#00
29 EB 39 HEX B#16#00
30 EB 40 HEX B#16#00
31 EB 41 HEX B#16#00
32 EB 42 HEX B#16#00
33 EB 43 HEX B#16#00
34 EB 44 HEX
@VAT Output Data -- Ex_6K_v1\SIMATIC 300 S...
Address Symbol Displ Status value Modify va
1 // Bit 0 of Byte 0 of OutputArea: TxBufferEmpty bit
2 A 10.0 BOOL true
3 // Byte 0 of Output Area: Control Field
4 AB 10 BIN 2#0000_0001
5 // Barcode counter
6 MD 200 DEC L#1
7 // Message Length
8 DB1.DBB 312 HEX B#16#04
9 // Local Buffer
10 DB1.DBB 16 HEX B#16#02
11 DB1.DBB 17 HEX B#16#18
12 DB1.DBB 18 HEX B#16#0D
13 DB1.DBB 19 HEX B#16#0A
14 DB1.DBB 20 HEX B#16#00
15 DB1.DBB 21 HEX B#16#00
16 DB1.DBB 22 HEX B#16#00
17 DB1.DBB 23 HEX B#16#00
18 DB1.DBB 24 HEX B#16#00
19 DB1.DBB 25 HEX B#16#00
20 DB1.DBB 26 HEX B#16#00
21 DB1.DBB 27 HEX B#16#00
22 DB1.DBB 28 HEX B#16#00
23 DB1.DBB 29 HEX B#16#00
24 DB1.DBB 30 HEX B#16#00
25 DB1.DBB 31 HEX B#16#00
26 DB1.DBB 32 HEX B#16#00
27 DB1.DBB 33 HEX B#16#00
28 DB1.DBB 34 HEX B#16#00
29 DB1.DBB 35 HEX B#16#00
30 DB1.DBB 36 HEX B#16#00
31 DB1.DBB 37 HEX B#16#00
32 DB1.DBB 38 HEX B#16#00
33 DB1.DBB 39 HEX B#16#00
34 DB1.DBB 40 HEX
Ex_6K_v1\SIMATIC 300 Station\...\S7 Program
RUN

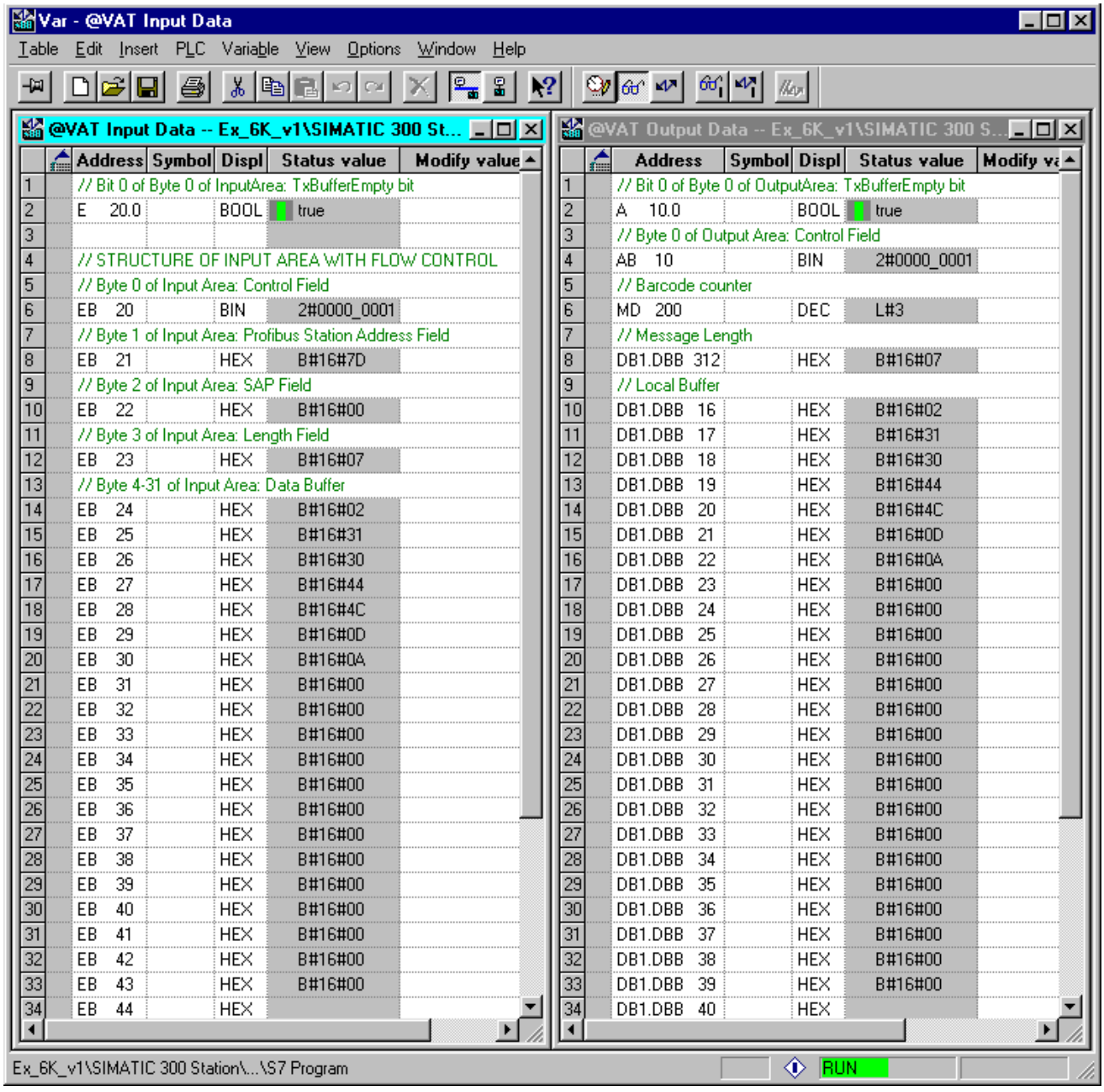
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Let's give a 2nd trigger to the scanner making another NoRead.
2nd message (NoRead) will be sent to the PLC.
Nothing seems to change but note Bit 0 toggling and barcode counter increasing.


Var - @VAT Input Data
Table Edit Insert PLC Variable View Options Window Help
@VAT Input Data -- Ex_6K_v1\SIMATIC 300 St...
Address Symbol Displ Status value Modify value
1 // Bit 0 of Byte 0 of InputArea: TxBufferEmpty bit
2 E 20.0 BOOL false
3
4 // STRUCTURE OF INPUT AREA WITH FLOW CONTROL
5 // Byte 0 of Input Area: Control Field
6 EB 20 BIN 2#0000_0000
7 // Byte 1 of Input Area: Profibus Station Address Field
8 EB 21 HEX B#16#7D
9 // Byte 2 of Input Area: SAP Field
10 EB 22 HEX B#16#00
11 // Byte 3 of Input Area: Length Field
12 EB 23 HEX B#16#04
13 // Byte 4-31 of Input Area: Data Buffer
14 EB 24 HEX B#16#02
15 EB 25 HEX B#16#18
16 EB 26 HEX B#16#0D
17 EB 27 HEX B#16#0A
18 EB 28 HEX B#16#00
19 EB 29 HEX B#16#00
20 EB 30 HEX B#16#00
21 EB 31 HEX B#16#00
22 EB 32 HEX B#16#00
23 EB 33 HEX B#16#00
24 EB 34 HEX B#16#00
25 EB 35 HEX B#16#00
26 EB 36 HEX B#16#00
27 EB 37 HEX B#16#00
28 EB 38 HEX B#16#00
29 EB 39 HEX B#16#00
30 EB 40 HEX B#16#00
31 EB 41 HEX B#16#00
32 EB 42 HEX B#16#00
33 EB 43 HEX B#16#00
34 EB 44 HEX
@VAT Output Data -- Ex_6K_v1\SIMATIC 300 S...
Address Symbol Displ Status value Modify va
1 // Bit 0 of Byte 0 of OutputArea: TxBufferEmpty bit
2 A 10.0 BOOL false
3 // Byte 0 of Output Area: Control Field
4 AB 10 BIN 2#0000_0000
5 // Barcode counter
6 MD 200 DEC L#2
7 // Message Length
8 DB1.DBB 312 HEX B#16#04
9 // Local Buffer
10 DB1.DBB 16 HEX B#16#02
11 DB1.DBB 17 HEX B#16#18
12 DB1.DBB 18 HEX B#16#0D
13 DB1.DBB 19 HEX B#16#0A
14 DB1.DBB 20 HEX B#16#00
15 DB1.DBB 21 HEX B#16#00
16 DB1.DBB 22 HEX B#16#00
17 DB1.DBB 23 HEX B#16#00
18 DB1.DBB 24 HEX B#16#00
19 DB1.DBB 25 HEX B#16#00
20 DB1.DBB 26 HEX B#16#00
21 DB1.DBB 27 HEX B#16#00
22 DB1.DBB 28 HEX B#16#00
23 DB1.DBB 29 HEX B#16#00
24 DB1.DBB 30 HEX B#16#00
25 DB1.DBB 31 HEX B#16#00
26 DB1.DBB 32 HEX B#16#00
27 DB1.DBB 33 HEX B#16#00
28 DB1.DBB 34 HEX B#16#00
29 DB1.DBB 35 HEX B#16#00
30 DB1.DBB 36 HEX B#16#00
31 DB1.DBB 37 HEX B#16#00
32 DB1.DBB 38 HEX B#16#00
33 DB1.DBB 39 HEX B#16#00
34 DB1.DBB 40 HEX
Ex_6K_v1\SIMATIC 300 Station\...\S7 Program
RUN

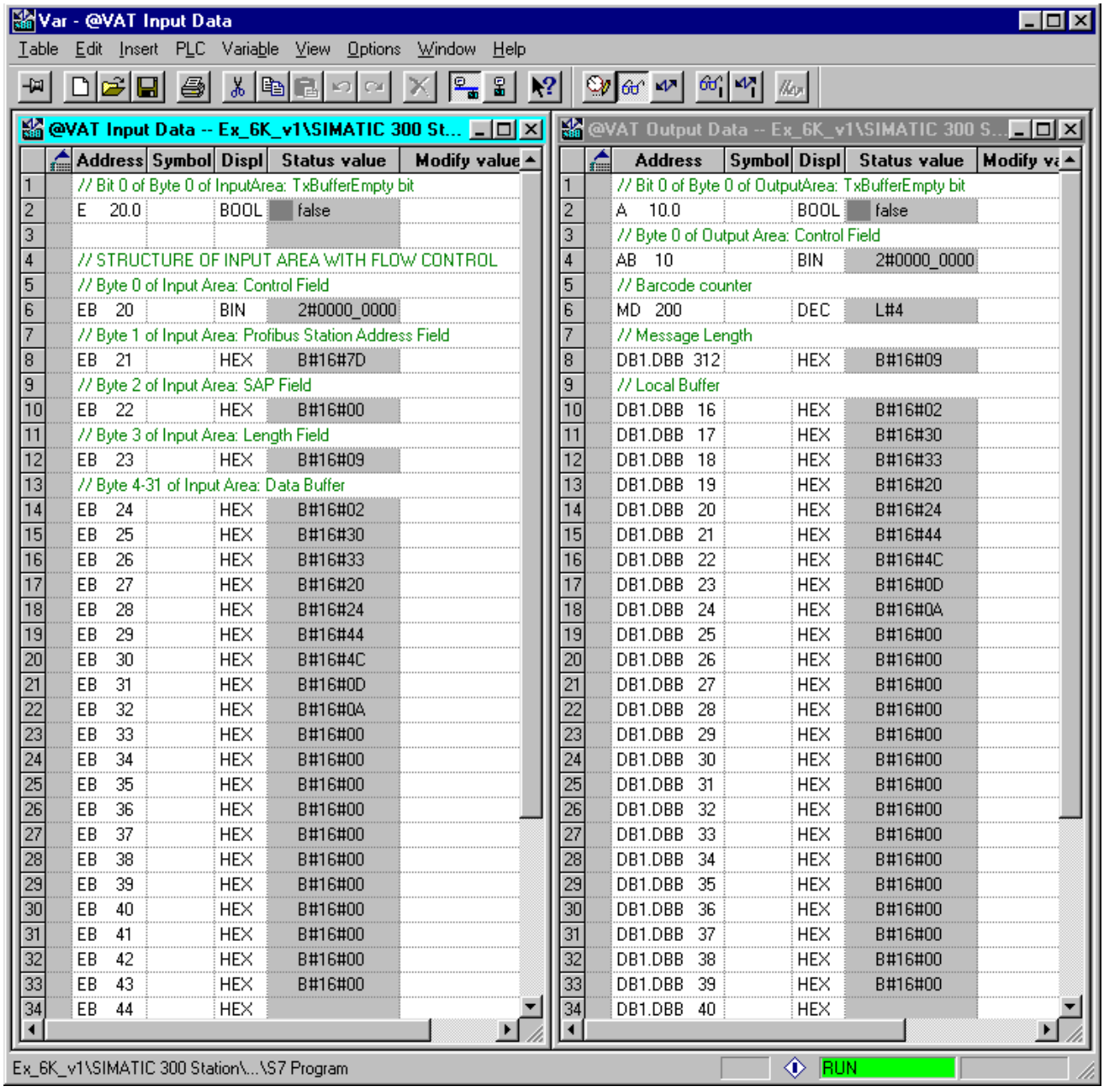
This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Let's give a 3rd trigger to the scanner reading "10DL" on the DL Barcode Test Chart.
3rd message ("10 DL") will be sent to the PLC.
Bit 0 toggles and barcode counter increases.


Var - @VAT Input Data
Table Edit Insert PLC Variable View Options Window Help
@VAT Input Data -- Ex_6K_v1\SIMATIC 300 St...
@VAT Output Data -- Ex_6K_v1\SIMATIC 300 S...
Address Symbol Displ Status value Modify value
// Bit 0 of Byte 0 of InputArea: TxBufferEmpty bit
E 20.0 BOOL true
// STRUCTURE OF INPUT AREA WITH FLOW CONTROL
// Byte 0 of Input Area: Control Field
EB 20 BIN 2#0000_0001
// Byte 1 of Input Area: Profibus Station Address Field
EB 21 HEX B#16#7D
// Byte 2 of Input Area: SAP Field
EB 22 HEX B#16#00
// Byte 3 of Input Area: Length Field
EB 23 HEX B#16#07
// Byte 4-31 of Input Area: Data Buffer
EB 24 HEX B#16#02
EB 25 HEX B#16#31
EB 26 HEX B#16#30
EB 27 HEX B#16#44
EB 28 HEX B#16#4C
EB 29 HEX B#16#0D
EB 30 HEX B#16#0A
EB 31 HEX B#16#00
EB 32 HEX B#16#00
EB 33 HEX B#16#00
EB 34 HEX B#16#00
EB 35 HEX B#16#00
EB 36 HEX B#16#00
EB 37 HEX B#16#00
EB 38 HEX B#16#00
EB 39 HEX B#16#00
EB 40 HEX B#16#00
EB 41 HEX B#16#00
EB 42 HEX B#16#00
EB 43 HEX B#16#00
EB 44 HEX
Address Symbol Displ Status value Modify va
// Bit 0 of Byte 0 of OutputArea: TxBufferEmpty bit
A 10.0 BOOL true
// Byte 0 of Output Area: Control Field
AB 10 BIN 2#0000_0001
// Barcode counter
MD 200 DEC L#3
// Message Length
DB1.DBB 312 HEX B#16#07
// Local Buffer
DB1.DBB 16 HEX B#16#02
DB1.DBB 17 HEX B#16#31
DB1.DBB 18 HEX B#16#30
DB1.DBB 19 HEX B#16#44
DB1.DBB 20 HEX B#16#4C
DB1.DBB 21 HEX B#16#0D
DB1.DBB 22 HEX B#16#0A
DB1.DBB 23 HEX B#16#00
DB1.DBB 24 HEX B#16#00
DB1.DBB 25 HEX B#16#00
DB1.DBB 26 HEX B#16#00
DB1.DBB 27 HEX B#16#00
DB1.DBB 28 HEX B#16#00
DB1.DBB 29 HEX B#16#00
DB1.DBB 30 HEX B#16#00
DB1.DBB 31 HEX B#16#00
DB1.DBB 32 HEX B#16#00
DB1.DBB 33 HEX B#16#00
DB1.DBB 34 HEX B#16#00
DB1.DBB 35 HEX B#16#00
DB1.DBB 36 HEX B#16#00
DB1.DBB 37 HEX B#16#00
DB1.DBB 38 HEX B#16#00
DB1.DBB 39 HEX B#16#00
DB1.DBB 40 HEX
Ex_6K_v1\SIMATIC 300 Station\...\S7 Program
RUN

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.

Let's give a 4th trigger to the scanner reading "03 $DL" on the DL Barcode Test Chart.
4th message ("03 $DL") will be sent to the PLC.
Bit 0 toggles and barcode counter increases.


Var - @VAT Input Data
Table Edit Insert PLC Variable View Options Window Help
@VAT Input Data -- Ex_6K_v1\SIMATIC 300 St...
Address Symbol Displ Status value Modify value
1 // Bit 0 of Byte 0 of InputArea: TxBufferEmpty bit
2 E 20.0 BOOL false
3
4 // STRUCTURE OF INPUT AREA WITH FLOW CONTROL
5 // Byte 0 of Input Area: Control Field
6 EB 20 BIN 2#0000_0000
7 // Byte 1 of Input Area: Profibus Station Address Field
8 EB 21 HEX B#16#7D
9 // Byte 2 of Input Area: SAP Field
10 EB 22 HEX B#16#00
11 // Byte 3 of Input Area: Length Field
12 EB 23 HEX B#16#09
13 // Byte 4-31 of Input Area: Data Buffer
14 EB 24 HEX B#16#02
15 EB 25 HEX B#16#30
16 EB 26 HEX B#16#33
17 EB 27 HEX B#16#20
18 EB 28 HEX B#16#24
19 EB 29 HEX B#16#44
20 EB 30 HEX B#16#4C
21 EB 31 HEX B#16#0D
22 EB 32 HEX B#16#0A
23 EB 33 HEX B#16#00
24 EB 34 HEX B#16#00
25 EB 35 HEX B#16#00
26 EB 36 HEX B#16#00
27 EB 37 HEX B#16#00
28 EB 38 HEX B#16#00
29 EB 39 HEX B#16#00
30 EB 40 HEX B#16#00
31 EB 41 HEX B#16#00
32 EB 42 HEX B#16#00
33 EB 43 HEX B#16#00
34 EB 44 HEX
@VAT Output Data -- Ex_6K_v1\SIMATIC 300 S...
Address Symbol Displ Status value Modify va
1 // Bit 0 of Byte 0 of OutputArea: TxBufferEmpty bit
2 A 10.0 BOOL false
3 // Byte 0 of Output Area: Control Field
4 AB 10 BIN 2#0000_0000
5 // Barcode counter
6 MD 200 DEC L#4
7 // Message Length
8 DB1.DBB 312 HEX B#16#09
9 // Local Buffer
10 DB1.DBB 16 HEX B#16#02
11 DB1.DBB 17 HEX B#16#30
12 DB1.DBB 18 HEX B#16#33
13 DB1.DBB 19 HEX B#16#20
14 DB1.DBB 20 HEX B#16#24
15 DB1.DBB 21 HEX B#16#44
16 DB1.DBB 22 HEX B#16#4C
17 DB1.DBB 23 HEX B#16#0D
18 DB1.DBB 24 HEX B#16#0A
19 DB1.DBB 25 HEX B#16#00
20 DB1.DBB 26 HEX B#16#00
21 DB1.DBB 27 HEX B#16#00
22 DB1.DBB 28 HEX B#16#00
23 DB1.DBB 29 HEX B#16#00
24 DB1.DBB 30 HEX B#16#00
25 DB1.DBB 31 HEX B#16#00
26 DB1.DBB 32 HEX B#16#00
27 DB1.DBB 33 HEX B#16#00
28 DB1.DBB 34 HEX B#16#00
29 DB1.DBB 35 HEX B#16#00
30 DB1.DBB 36 HEX B#16#00
31 DB1.DBB 37 HEX B#16#00
32 DB1.DBB 38 HEX B#16#00
33 DB1.DBB 39 HEX B#16#00
34 DB1.DBB 40 HEX
Ex_6K_v1\SIMATIC 300 Station\...\S7 Program
RUN

Procedure cyclically continues...

This documentation is property of DATALOGIC.
Reproduction in whole or in part is forbidden without the written consent of DATALOGIC.